

03-06-06

#AF
Jew



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Re Application of)
Richard Francis Russell, et al.) Group: 2142
Serial No.: 09/957,014)
Filed: September 20, 2001)
Title: AUTOMATIC REMOTE ASSIGNMENT OF INTERNET)
PROTOCOL ADDRESS INFORMATION TO A NETWORK) Examiner: B. Prieto
DEVICE)

LETTER

MS APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Enclosed herewith is the Brief of Appellant in the above-identified patent application.

The \$500.00 fee is enclosed.

In the event Applicants have overlooked the need for an extension of time, an additional extension of time, payment of fee, or additional payment of fee, Applicants hereby conditionally petition therefor and authorizes that any charges be made to Deposit Account No. 20-0095, TAYLOR & AUST, P.C.

Respectfully submitted,

Paul C. Gosnell
Registration No. 46,735

Attorney for Appellants

"EXPRESS MAIL" Mailing Number EV 620802603 US

Date of Deposit March 3, 2006

I hereby certify that this paper or fee is being deposited with the United States Postal Service "EXPRESS MAIL POST OFFICE TO ADDRESSEE" service under 37 CFR 1.10 on the date indicated above and is addressed to MS APPEAL BRIEF - PATENTS Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450

Paul C. Gosnell, Reg. No. 46,735

(Typed Name of Person Mailing Paper or Fee)

(Signature of Person Mailing Paper or Fee)

PCG14/ts

TAYLOR & AUST, P.C.
12029 E. Washington Street
Indianapolis, IN 46229
Telephone: 317-894-0801
Facsimile: 317-894-0803

Encs.: Return postcard
Check No. 14017



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of)	
Richard Francis Russell, et al.)	Group: 2142
Serial No.: 09/957,014)	
Filed: September 20, 2001)	
Title: AUTOMATIC REMOTE ASSIGNMENT OF INTERNET)	
PROTOCOL ADDRESS INFORMATION TO A NETWORK)	Examiner: B. Prieto
DEVICE)	

BRIEF OF APPELLANT

MS APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This appeal is taken from the decision of the Examiner, dated October 4, 2005, finally rejecting claims 1-25, all of the claims that are under consideration in the above-captioned patent application. Appellants timely filed a Notice of Appeal in this matter on January 3, 2006.

03/07/2006 MBIZUNES 00000126 09957014

01 FC:1402

500.00 OP

I. TABLE OF CONTENTS

Real Party In Interest	Page 3
Related Appeals and Interferences.....	Page 4
Status of Claims	Page 5
Status of Amendments	Page 6
Summary of Claimed Subject Matter	Page 7
Grounds of Rejection To Be Reviewed On Appeal.....	Page 11
Argument	Page 12
Claims Appendix	Page 47
Evidence Appendix.....	Page 52
Related Proceedings Appendix	Page 53

II. REAL PARTY IN INTEREST

The real party in interest in this appeal is Lexmark International, Inc., a corporation organized and existing under the laws of the State of Delaware, which owns the entire interest in this patent application as set forth in the underlying claimed invention.

III. RELATED APPEALS AND INTERFERENCES

No related Appeals or Interferences are known to the Appellants.

IV. STATUS OF CLAIMS

Pending: 1-25.

Canceled: None

Allowed: None.

Objected To: None.

Rejected: 1-25.

Withdrawn from Consideration: None.

On Appeal: 1-25.

V. STATUS OF AMENDMENTS

A Reply Under 37 CFR 1.116 was submitted in this case on August 17, 2005, in response to the final rejection in the Office Action mailed June 17, 2005. The Reply did not include any claim amendments. The Reply was not entered, as indicated in the Advisory Action mailed September 14, 2005 and as discussed in an Interview with the Examiner on September 12, 2005. A Request For Continued Examination was filed on September 13, 2005, and the claims were finally rejected in the Office Action mailed October 4, 2005.

VI. SUMMARY OF CLAIMED SUBJECT MATTER

The present invention generally relates to assignment of internet protocol addresses and, more particularly, to automatically assigning internet protocol address information to a network device, such as a low-cost network adapter.

Referring to Fig. 1 there is there is shown networked imaging system 10 that includes a computer 12, a networked device 14 and a network 16. Computer 12 includes software identified as a printer driver 18 and an operating system 20. Printer driver 18 and operating system 20 are communicatively interconnected. Networked device 14 may be an imaging device, such as a printer. In the embodiment of the invention described, networked device 14 will be in the form of a printer. Networked device 14 includes printer firmware 22 and a low-cost network adapter (LCNA) 24, which are communicatively interconnected. All network traffic directed to networked device 14 flows through LCNA 24 to printer firmware 22. Printer firmware 22 is responsible for generating a printed page on networked device 14, and printer firmware 22 relies on LCNA 24 to deliver printer control information and print data thereto. Network 16, such as a LAN, provides communicative interconnection between computer 12 and networked device 14 and other devices connected thereto which may or may not contain LCNAs. (Spec. at page 3, lines 12-28).

Printer driver 18 includes a data generation component 26, a printer driver user interface 28 and low-cost network adapter (LCNA) host software 30. Printer driver 18 contains the algorithms for assigning IP addresses, and more particularly, for automatically assigning an IP address to LCNA 24. Data generation component 26 generates data to be sent to networked device 14. (Spec at page 3, line 31 to page 4, line 2).

LCNA host software 30 communicates with IP stack 34 to obtain the IP address for networked device 14. If no IP address is available for networked device 14, then LCNA host software 30 is responsible for discovering LCNA 24 equipped devices on network 16. LCNA host software 30 configures LCNA 24 equipped devices, when appropriate, and provides a print connection over which data can be sent to networked device 14 through LCNA 24. (Spec at page 4, lines 22-27).

LCNA 24 does not contain a mechanism for obtaining an IP address. Therefore, LCNA 24 depends on the operation of LCNA host software 30 on computer 12 to provide IP information thereto. LCNA 24 may be implemented as an application specific integrated circuit (ASIC). (Spec at page 4, lines 28-31).

In one embodiment, a method of automatically assigning an internet protocol address to a device 14 includes providing a network 16, providing a computer 12 communicatively coupled to the network 16; providing a network adapter LCNA 24 to communicatively couple device 14 to network 16, network 16 providing communicative interconnection between computer 12 and network adapter LCNA 24.

Referring now to Fig. 2, in the present embodiment, computer 12 performs the steps of generating an internet protocol address (Step S108; Spec at page 6, lines 10-13); incorporating the internet protocol address in an address resolution protocol probe; and sending the address resolution protocol probe on network 16 (Step S110; Spec at page 6, lines 13-15); and determining whether a response to the address resolution protocol probe indicates that the internet protocol address is in use (Step S112; Spec at page 6, lines 15-17), wherein if the internet protocol address is not in use, then performing the step of assigning

the internet protocol address to the network adapter LCNA 24 via network 16 (Step S118; Spec at page 6, lines 17-18 and page 7, lines 18-24).

Another embodiment of the present invention is a method of automatically assigning an internet protocol address to a device 14. The method includes providing a network 16; providing a computer 12 communicatively coupled to network 16; and providing a low-cost network adapter LCNA 24 to communicatively couple device 14 to network 16, network 16 providing communicative interconnection between computer 12 and low-cost network adapter LCNA 24.

Referring now to Fig. 2, in the present embodiment, computer 12 performs the steps of broadcasting a discovery packet on network 16 (Step S100, Spec at page 5, lines 13-18); receiving a response from low-cost network adapter LCNA 24 (Step S102, Spec at page 5, lines 18-24); determining if low-cost network adapter LCNA 24 has a valid internet protocol address (Step S104, Spec at page 5, lines 25-32).

If low-cost network adapter LCNA 24 does not have a valid internet protocol address, computer 12 then performs the steps of generating an internet protocol address (Step S108; Spec at page 6, lines 10-13); incorporating the internet protocol address in an address resolution protocol probe; and sending the address resolution protocol probe on network 16 (Step S110; Spec at page 6, lines 13-15); and determining whether a response to the address resolution protocol probe indicates that the internet protocol address is in use (Step S112; Spec at page 6, lines 15-17), wherein if the internet protocol address is not in use, then performing the step of assigning the internet protocol address to low-cost network adapter LCNA 24 via network 16 (Step S118; Spec at page 6, lines 17-18 and page 7, lines 18-24).

Yet another embodiment of the present invention is directed to a network based imaging system 10. Imaging system 10 includes a network 16; a computer 12 communicatively coupled to network 16; an imaging device 14; and a network adapter LCNA 24 communicatively coupling imaging device 14 to network 16, network 16 providing communicative interconnection between computer 12 and network adapter 16.

Referring now to Fig. 2, in the present embodiment, computer 12 executes instructions which generate an internet protocol address (Step S108; Spec at page 6, lines 10-13), incorporate the internet protocol address into an address resolution protocol probe, and send the address resolution protocol probe on network 16 (Step S110; Spec at page 6, lines 13-15), utilize a response to the address resolution protocol probe to determine if the internet protocol address is in use (Step S112; Spec at page 6, lines 15-17), and if the internet protocol address is not in use, then assign the internet protocol address to network adapter LCNA 24 via network 16 (Step S118; Spec at page 6, lines 17-18 and page 7, lines 18-24).

VII. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. Claims 1, 9-10, 17, and 25 were rejected under 35 U.S.C. §103(a) as being obvious over Buse, et al., U.S. Patent No. 6,810,420 B1 in view of Cheshire, S., Current Meeting Report, Cheshire, et al., 03/99.

B. Claims 2-6 and 18-22 were rejected under 35 U.S.C. §103(a) as being obvious over Buse in view of Cheshire, and in further view of Reed, et al., U.S. Patent No. 6,061,739.

C. Claims 7, 11-16, and 23 were rejected under 35 U.S.C. §103(a) as being unpatentable over Buse in view of Cheshire, and in further view of Mellquist, U.S. Patent No. 6,115,545.

D. Claims 8 and 24 were rejected under 35 USC §103(a) as being unpatentable over Buse in view of Cheshire, in further view of Mellquist, and in further view of Troll, Request for Comments: 2563, May 1999, Troll R.

VIII. ARGUMENT

A. CLAIMS 1, 9-10, 17, AND 25 ARE PATENTABLE UNDER 35 U.S.C. 103(a)

In the Final Office Action dated October 4, 2005, claims 1, 9, 10, 17, and 25 were rejected under 35 U.S.C. §103(a) as being unpatentable over Buse, et al., U.S. Patent No. 6,810,420 B1 (hereinafter, Buse) in view of Cheshire, S., Current Meeting Report, Cheshire, et al., 03/99 (hereinafter, Cheshire).

However, in determining whether obviousness is established by combining the teachings of the prior art, “the test is what the combined teachings of the references would have suggested to those of ordinary skill in the art,” and the combined teachings of the prior art references must suggest, expressly or by implication, the improvements embodied by the invention. *In re GPAC Inc.* 35 USPQ2d 1116, 1123 (Fed Cir. 1995).

However, as set forth below, Appellants submit that claims 1, 9, 10, 17, and 25 are not disclosed, taught, or suggested by Buse in view of Cheshire, and are therefore patentable in their present form.

1. BUSE

Buse discloses a discovery scheme which can be operated by a proxy device such as a personal computer coupled to a local area network, and which facilitates the discovery of devices which may or may not be configured with an IP address (col. 1, lines 39-42). The discovery protocol performed by the proxy employs three basic packets, and when the proxy has resolved an IP address for the device, it sends an IP address allocated to the device, which then configures itself with the supplied parameters, and sends an “I_AM_HERE” frame with the address field being set to the allocated IP address (col. 2, lines 22-58, Figs. 2 and 3).

In order to resolve an IP address for the device, the proxy first sends a DHCP request, and if a DHCP server is available, that server provides a DHCP response including an IP address (col. 3, lines 23-26). If there is no DHCP response, the proxy allocates an IP address using Automatic Private IP addressing (col. 3, lines 28-37), and may verify that there is no address conflict using address resolution protocol or an ICMP echo request (col. 3, lines 37-41).

Thus, Buse provides to a device an IP address obtained via either a DHCP request or Automatic Private IP addressing.

2. CHESHIRE

Cheshire discloses automatic IP address assignment for a link local address with IPv4 (page 1), specifically the IPv4 self-configuration as currently implemented by Apple and MS (bottom paragraph of page 2). Operation as implemented in Mac OS 8.5 includes using a DHCP discover, and if no DHCP server is discovered, picking a random address, sending an ARP probe to verify that the address is not already in use, and if the address is in use, iterating the picking and repeating steps 10 times at most, otherwise configuring the computer's interface with the IP address (page 3).

Thus, Cheshire discloses a computer obtaining for itself an IP address via either a DHCP discover request or by picking a random address, verifying that it is not in use via an APR probe, and configuring that computer's interface with the IP address.

3. CLAIM 1 IS PATENTABLE OVER BUSE IN VIEW OF CHESHIRE

Appellants' claim 1 is directed to a method of automatically assigning an internet protocol address to a device. Claim 1 recites, in part, providing a network; providing a computer communicatively coupled to said network; providing a network adapter to

communicatively couple said device to said network, said network providing communicative interconnection between said computer and said network adapter.

Claim 1 also recites said computer performing the steps of: generating an internet protocol address; incorporating said internet protocol address in an address resolution protocol probe; sending said address resolution protocol probe on said network; and determining whether a response to said address resolution protocol probe indicates that said internet protocol address is in use; wherein if said internet protocol address is not in use, then performing the step of assigning said internet protocol address to said network adapter via said network.

In contrast to claim 1, Buse discloses (1) obtaining an IP address via either a DHCP request or Automatic Private IP addressing, wherein (2) it may be verified that there is no address conflict using address resolution protocol or an ICMP echo request (col. 3, lines 37-41), and (3) providing the device with an address (col. 3, lines 23-41).

Although the Buse invention may arguably provide a device with an IP address, it does not do so by (1) generating an IP address; (2) incorporating the IP address in an ARP probe; (3) sending the ARP probe on the network; (4) determining whether a response to the ARP probe indicates that the IP address is in use; and (5), assigning the IP address to the network adapter via the network if the internet protocol address is not in use, as recited in claim 1.

Accordingly, assuming arguendo that Buse does provide a device with an IP address, Buse does not do so in a manner as recited in claim 1. Rather, the Buse approach is distinctly different from Appellants' invention of claim 1; the steps taken in the Buse disclosure are not common to claim 1, and clearly do not disclose, teach, or suggest

generating an internet protocol address; incorporating the internet protocol address in an address resolution protocol probe; sending the address resolution protocol probe on the network; and determining whether a response to the address resolution protocol probe indicates that the internet protocol address is in use; wherein if the internet protocol address is not in use, then performing the step of assigning the internet protocol address to the network adapter via the network, as recited in claim 1.

Appellants respectfully submit that the Patent and Trademark Office determines the scope of claims in patent applications not solely on the basis of the claim language, but upon giving claims their broadest reasonable construction “in light of the specification as it would be interpreted by one of ordinary skill in the art.” *In re Am. Acad. of Sci. Tech. Ctr.*, 367 F.3d 1359, 1364 [70 USPQ2d 1827] (Fed. Cir. 2004) (Emphasis added).

Appellants respectfully direct the Board’s attention to Appellants’ specification at page 1, line 29 to page 2, line 14, which is reproduced as follows:

There are several industry standards by which a network device can automatically obtain an IP address information. Such standards include the aforementioned DHCP, Universal Plug and Play (UPnP) and other forms of Automatic Private IP Addressing (APIPA). Each of these standards require that significant network transactions be initiated and conducted by the network device itself which requires hardware and configuration storage, making them cost prohibitive for low-cost devices.

What is needed in the art is an apparatus and a method by which a device on a computer network can be assigned an IP address automatically, without the overhead of supporting traditional address assignment protocols.

The present invention provides an apparatus and a method by which a device on a computer network can be assigned an IP address automatically, without the overhead of supporting the traditional address assignment protocols, such as DHCP, within the devices themselves. (Emphasis added).

Thus, Appellants claims may not be properly interpreted to encompass a method of obtaining an IP address via either a DHCP request or Automatic Private IP addressing, as disclosed by Buse.

Rather, Appellants' claimed invention is directed to a method by which a device on a computer network can be assigned an IP address automatically, without the overhead of supporting the traditional address assignment protocols, such as DHCP, within the devices themselves.

The Examiner acknowledges that Buse does not disclose, teach, or suggest the use of an ARP probe "nor where assigning an IP address to a device is performed by assigning the IP address to the network adapter of the device which connects the device to the network (Page 3 of Office Action mailed 10/4/05).

Rather, the Examiner relies upon Cheshire for as disclosing "where assigning an IP address to a device is performed by assigning the IP address to the network adapter of the device which connects the device to the network."

However, Cheshire simply does not make up for the deficiency of the Buse disclosure in rejecting claim 1.

For example, Cheshire discloses using a DHCP discover, and if no DHCP server is discovered, picking a random address, sending an ARP probe to verify that the address is not already in use, and if the address is in use, iterating the picking and repeating steps 10 times at most, otherwise configuring the computer's interface with the IP address (page 3).

However, the Cheshire disclosure specifically pertains to self-configuration (see bottom paragraph of page 2 of 7 of the Cheshire disclosure).

In addition, as set forth above, Appellants' claimed invention does not encompass a DHCP self-configuration approach to IP addressing.

Presently, Appellants' call the Board's attention to the fact that claim 1 recites that the (1) generating an IP address; (2) incorporating the IP address in an ARP probe; (3) sending the ARP probe on the network; (4) determining whether a response to the ARP probe indicates that the IP address is in use; and (5), assigning the IP address to the network adapter via the network if the internet protocol address is not in use, are performed relative to a network adapter that communicatively couples the device to the network, the network providing communicative interconnection between the computer and the network adapter, as recited in claim 1.

Thus, the network adapter of claim 1 is not associated with the computer that performs the assigning of the IP address, but rather, is in communication with the computer via the network.

In contrast, Cheshire is explicitly directed to self-configuration, which is known in the art as being a device that configures itself. In the case of Cheshire, one skilled in the art would clearly recognize that the steps taken by Cheshire are those steps taken in the MAC OS 8.5 for a computer to configure its own interface with an IP address, which is made explicit in the Cheshire disclosure on page 3 of 7.

Thus, Cheshire teaches self-configuration, as opposed to providing an IP address for another device, separate and distinct from the computer that obtains the IP address, which is connected via a network to the computer that obtains the IP address.

Accordingly, since Cheshire does not overcome the deficiency of Buse, as applied to claim 1, Buse and Cheshire, taken alone or in combination, do not disclose, teach, or suggest the subject matter of claim 1.

Although the Examiner asserts in the Response to Arguments that arguments against a reference individually cannot show nonobviousness where the rejections are based on a combination of references, Appellants respectfully submit that, as set forth above, the combination of Buse and Cheshire would not yield Appellants' claimed invention, since all of the limitations of claim 1 are not taught, disclosed, or suggested by Buse and Cheshire.

MPEP 2142 provides that to establish a prima facie case of obviousness, the prior art reference (or references when combined) must teach or suggest all the claim limitations.

Since Buse and Cheshire, taken alone or in combination, do not disclose, teach, or suggest all the limitations of claim 1, a prima facie case of obviousness has not been established against claim 1, and hence, claim 1 is allowable over Buse in view of Cheshire.

Notwithstanding the above, it would not be obvious to combine the teachings of Buse and Cheshire to achieve Appellants' invention of claim 1, i.e., to combine self-configuration, as disclosed by Cheshire, with the allocation of IP addresses by proxy, as disclosed by Buse, in order to achieve Appellants' claimed invention, at least for the reason that there would be no motivation to modify Buse with Cheshire, since each is a different approach in obtaining IP addresses.

Although the Examiner asserts that the motivation may be found in either the references themselves or knowledge generally available to one of ordinary skill in the art, relying upon *In re Fine*, 5 USPQ2d 1596 (Fed. Cir. 1988), the Examiner has not provided particular findings as to the reason the skilled artisan, with no knowledge of the claimed

invention, would have selected the particular components from Cheshire and Buse for combination in the manner claimed.

There was no specific understanding or principle within the knowledge of a skilled artisan that would have motivated one with no knowledge of Appellants' invention to make the combination in the manner of claim 1. *In re Kotzab*, 55 USPQ2d 1313, 1317 (Fed. Cir. 2000). See also *In Re Lee*, 277 F.3d 1338 (Fed. Cir. 2002).

In addition, prior art references in combination do not make the invention obvious unless something in the prior art references would suggest an advantage to be derived from combining their teachings. *In re Sernaker*, 217 USPQ 1 (Fed. Cir. 1983). See also *In re GPAC*, 35 USPQ2d 1116, 1123 (Fed. Cir. 1995).

Also, MPEP 2144 provides that the expectation of some advantage is the strongest rationale for combining references.

However, neither Buse nor Cheshire disclose, teach, or suggest an advantage to be derived from combining their teachings in the manner attempted in rejecting claim 1.

Further, Appellants submit that the asserted combination is based on impermissible hindsight reconstruction. It is impermissible to use the claimed invention as an instruction manual or "template" to piece together the teachings of the prior art so that the claimed invention is rendered obvious. *In re Fritch*, 23 USPQ2d 1780, 1784 (Fed. Cir. 1992).

Buse is directed to configuration by proxy, whereas Cheshire is directed to self-configuration.

Since, as set forth above, there is no advantage disclosed, taught, or suggested by either of the references to modify Buse with Cheshire, and no asserted principle within the knowledge of a skilled artisan that would have motivated one with no knowledge of

Appellants' invention to make the combination, it seems clear that hindsight reconstruction has been employed in rejecting Appellants' claims.

Accordingly, claim 1 is not obvious over Buse in view of Cheshire.

Thus, for at least the reasons set forth above, Appellants submit that claim 1 is patentable in its present form.

Accordingly, Appellants respectfully request that the Board reverse the rejection of claim 1 under 35 U.S.C. 103(a).

4. CLAIMS 9 AND 10 ARE PATENTABLE OVER BUSE IN VIEW OF CHESHIRE

Claims 9 and 10 are believed allowable due to their dependence on otherwise allowable base claim 1. In addition, claims 9 and 10 further and patentably define Appellants' invention over Buse in view of Cheshire.

Thus, for at least the reasons set forth above, Appellants submit that claims 9 and 10 are patentable in its present form.

Accordingly, Appellants respectfully request that the Board reverse the rejection of claims 9 and 10 under 35 U.S.C. 103(a).

5. CLAIM 17 IS PATENTABLE OVER BUSE IN VIEW OF CHESHIRE

Appellants' claim 17 is directed to a network based imaging system. Claim 17 recites, in part, wherein said computer executes instructions which generate an internet protocol address, incorporate said internet protocol address into an address resolution protocol probe, send said address resolution protocol probe on said network, utilize a response to said address resolution protocol probe to determine if said internet protocol

address is in use and if said internet protocol address is not in use, then assign said internet protocol address to said network adapter via said network.

Claim 17 is believed allowable for substantially the same reasons as set forth above with respect to claim 1.

Claim 17 also recites, in part, an imaging device; and a network adapter communicatively coupling said imaging device to said network, said network providing communicative interconnection between said computer and said network adapter.

Buse simply does not disclose, teach, or suggest an imaging device, and nor does the Examiner assert as much. In addition, although Cheshire offhandedly mentions “printers,” (page 5), Cheshire does not disclose, teach, or suggest an imaging device; and a network adapter communicatively coupling the imaging device to the network, the network providing communicative interconnection between the computer and the network adapter.

MPEP 2142 requires that in order to establish a *prima facie* case of obviousness, all claim limitations must be taught or suggested by the prior art references.

Because Buse in view of Cheshire simply do not disclose, teach, or suggest all of the limitations of claim 17, e.g., an imaging device; and a network adapter communicatively coupling the imaging device to the network, the network providing communicative interconnection between the computer and the network adapter, Appellants invention of claim 17 is not obvious over Buse in view of Cheshire as per MPEP2142.

In rejecting claim 17, the Examiner asserts that “in the absence of an express intent to impart a novel meaning to the claim terms, the words are presumed to take on the ordinary and customary meanings attributed to them by one of ordinary skill in the art.”

Further, the Examiner asserts that “The broadest reasonable interpretation has been applied to the claims as mandated, thereby, claimed ‘imaging device’, for the purposes of examination given the broadest reasonable interpretation is a device.” (Emphasis added).

However, the Patent and Trademark Office determines the scope of claims in patent applications not solely on the basis of the claim language, but upon giving claims their broadest reasonable construction “in light of the specification as it would be interpreted by one of ordinary skill in the art.” *In re Am. Acad. of Sci. Tech. Ctr.*, 367 F.3d 1359, 1364 [70 USPQ2d 1827] (Fed. Cir. 2004) (Emphasis added).

Appellants’ specification provides that network device 14 may be an imaging device, such as a printer (p.3, l.20). In interpreting “imaging device” to include any “device” generally, the Examiner has truncated Appellants’ chosen term, and simply removed the word, “imaging,” along with the meaning that “imaging” imports to “imaging device.” Interpreted in light of Appellants’ specification, one of ordinary skill in the art would not interpret “imaging device” broadly enough to encompass any “device.” Rather, an imaging device is known in the art to pertain to, e.g., a printer, a copier, an all-in-one unit that combines printing, copying, and faxing, etc., and the like.

Accordingly, Appellants respectfully submit that upon giving claim 1 its broadest reasonable construction in light of the specification as it would be interpreted by one of ordinary skill in the art, the claim 1 term, “imaging device,” would not be interpreted by one skilled in the art as broadly as the general term, “device,” as asserted by the Examiner.

Since Buse and Cheshire, taken alone or in combination, do not disclose, teach, or suggest all the limitations of claim 17, a case of prima facie obviousness has not been

established against claim 17, and hence, claim 17 is allowable over Buse in view of Cheshire.

In addition, it would not be obvious to combine Buse and Cheshire for substantially the same reasons as set forth above with respect to claim 1.

Accordingly, for at least the reasons set forth above, Appellants respectfully submit that claim 17 is patentable over the cited references, Buse in view of Cheshire.

Thus, for at least the reasons set forth above, Appellants submit that claim 17 is patentable in its present form.

Accordingly, Appellants respectfully request that the Board reverse the rejection of claim 17 under 35 U.S.C. 103(a).

6. CLAIM 25 IS PATENTABLE OVER BUSE IN VIEW OF CHESHIRE

Claim 25 is believed allowable due to its dependence on otherwise allowable base claim 17.

Thus, for at least the reasons set forth above, Appellants submit that claim 25 is patentable in its present form.

Accordingly, Appellants respectfully request that the Board reverse the rejection of claim 25 under 35 U.S.C. 103(a).

B. CLAIMS 2-6 AND 18-22 ARE PATENTABLE UNDER 35 U.S.C. 103(a)

In the Final Office Action dated October 4, 2005, claims 2-6 and 18-22 were rejected under 35 U.S.C. §103(a) as being unpatentable over Buse in view of Cheshire, and in further view of Reed, et al., U.S. Patent No. 6,061,739 (hereinafter, Reed).

However, in determining whether obviousness is established by combining the teachings of the prior art, “the test is what the combined teachings of the references would

have suggested to those of ordinary skill in the art,” and the combined teachings of the prior art references must suggest, expressly or by implication, the improvements embodied by the invention. *In re GPAC Inc.* 35 USPQ2d 1116, 1123 (Fed Cir. 1995).

However, as set forth below, Appellants submit that claims 2-6 and 18-22 are not disclosed, taught, or suggested by Buse in view of Cheshire, and in further view of Reed, and are therefore patentable in their present form.

1. BUSE

Buse is summarized at the beginning of section VIII(A)(1) of this Brief, which for the sake of brevity is not repeated here.

2. CHESHIRE

Cheshire is summarized at the beginning of section VIII(A)(2) of this Brief, which for the sake of brevity is not repeated here.

3. REED

Reed discloses a method for assigning a network address to a new device coupled to a network without any additional infrastructure or pre-existing knowledge of the hardware address of the device (col. 4, lines 19-22). The device attempts to establish a connection on the network, resulting in ARP requests being generated (col. 4, lines 22-25). The device monitors the communications on the network for unanswered ARP requests (col. 4, lines 25-27). When the device sees N unanswered ARP requests in a given length of time, it adopts the requested network address and responds to the ARP with its hardware address (col. 4, lines 27-30, Fig. 2).

Thus, the Reed device performs self-configuration.

**4. CLAIMS 2-6 ARE PATENTABLE OVER BUSE IN VIEW OF CHESHIRE,
AND IN FURTHER VIEW OF REED**

Each of claims 2-6 depend directly or indirectly from claim 1. As set forth above with respect to claim 1, Buse and Cheshire, taken alone or in combination, would not yield the subject matter of claim 1, and the subject matter of claim 1 is not obvious over Buse in view of Cheshire.

Appellants respectfully submit that Reed does not overcome the deficiency of Buse in view of Cheshire, as applied to claim 1, nor does the Examiner assert as much.

In rejecting claim 1, the Examiner acknowledges that Buse does not disclose, teach, or suggest the use of an ARP probe, “nor where assigning an IP address to a device is performed by assigning the IP address to the network adapter of the device which connects the device to the network (Page 3 of Office Action mailed 10/4/05).

Rather, the Examiner relies upon Cheshire for as disclosing “where assigning an IP address to a device is performed by assigning the IP address to the network adapter of the device which connects the device to the network.”

However, as set forth above with respect to claim 1, Cheshire simply does not make up for the deficiency of the Buse disclosure as applied to claim 1.

In addition, Reed does not make up for the deficiency of Buse and Cheshire as applied to claim 1.

Like Cheshire, Reed discloses self-configuration of an IP address.

For example, Reed et al, discloses that the device attempts to establish a connection, causing ARP requests to be generated, and when the device sees N unanswered ARP requests (where N is a preset threshold) in a given length of time, the device adopts the

requested network address and responds to the ARP with its hardware address (col. 4, lines 22-30, Fig. 2).

Thus, by sending and responding to ARP communications, the device configures itself with an IP address, in contrast to claim 1, wherein a computer performs the step of assigning the internet protocol address to the network adapter that communicatively couples the device to the network, via the network, i.e., the network adapter for the device is configured by the computer via the network.

Thus, since Reed does not overcome the deficiency of Buse and Cheshire as applied to claim 1, the combination of Buse, Cheshire, and Reed would not yield Appellants' claimed invention.

Although the Examiner asserts in the Response to Arguments that arguments against a reference individually cannot show nonobviousness where the rejections are based on a combination of references, Appellants respectfully submit that, as set forth above, the combination of Buse, Cheshire, and Reed would not yield Appellants' claimed invention, since all of the limitations of claim 1 are not taught, disclosed, or suggested by Buse, Cheshire, and Reed, taken alone or in combination.

MPEP 2142 provides that to establish a prima facie case of obviousness the prior art reference (or references when combined) must teach or suggest all the claim limitations.

Since Buse, Cheshire, and Reed, taken alone or in combination, do not disclose, teach, or suggest all the limitations of claim 1, claim 1 a case of prima facie obviousness has not been established against claim 1, and hence, claim 1 is allowable over Buse in view of Cheshire and in further view of Reed.

Claims 2-6 are thus believed allowable due to their dependence, directly or indirectly, on otherwise allowable base claim 1.

Notwithstanding the above, since Reed is directed to and discloses self-configuration, for substantially the same reasons as set forth above with respect to claim 1 regarding Buse in view of Cheshire, it would not have been obvious to combine the teachings of Cheshire and Reed, i.e., self-configuration, with the allocation of IP addresses by proxy, as disclosed by Buse.

For example, the mere fact that the prior art could be so modified would not have made the modification obvious unless the prior art suggested the desirability of the modification. *In re Laskowski*, 10 USPQ2d 1397 (Fed. Cir. 1989). In addition, prior art references in combination do not make the invention obvious unless something in prior art references would suggest advantage to be derived from combining their teachings. *In re Sernaker*, 217 USPQ 1 (Fed. Cir. 1983).

Also, MPEP 2144 provides that the expectation of some advantage is the strongest rationale for combining references.

However, there is nothing disclosed, taught, or suggested in either of the Buse or Cheshire or Reed references that would suggest the desirability of the asserted combination or that there would be an advantage to be derived from their teachings. Thus, it would not have been obvious to combine the prior art references since there is simply nothing in those references suggests that their teachings could be successfully combined to yield advantageous results in the primary reference. *In re Sernaker*, 217 USPQ 1 (Fed. Cir. 1983).

In addition, Appellants submit that since Cheshire and Reed are directed to self-configuration, in contrast to Buse, one would not have been motivated to modify Buse with Cheshire and Reed, since Buse is directed to configuration by proxy, and also discloses what is purported to be an operational method to configure a device by proxy, that does not purport to need modification in order to achieve its desired result.

That is, there would be no motivation to modify a method for configuration by proxy that is operational in of itself (Buse), much less by incorporating aspects of a method for self-configuration (Cheshire and/or Reed).

Appellants further contend that the way in which the Examiner has assembled the combination of Buse, Cheshire and Reed, without any advantage disclosed, taught, or suggested by the references, is tantamount to impermissible hindsight reconstruction of Appellants' claims.

It is impermissible to use the claimed invention as an instruction manual or "template" to piece together the teachings of the prior art so that the claimed invention is rendered obvious. In re Fritch, (CA FC) 23 USPQ2d 1780, 1784 (Fed. Cir. 1992). For example, Buse is directed to configuration by proxy, whereas Cheshire and Reed are directed to self-configuration.

Since there is no advantage disclosed, taught, or suggested by any of the references (Buse, Cheshire, and/or Reed) to modify Buse with Cheshire or Reed, it follows that there would not be a motivation to combine the references, and consequently, Appellants submit that the asserted combination is based on impermissible hindsight reconstruction.

Accordingly, Appellants respectfully submit that it would not be obvious to modify Buse with Cheshire and Reed, and thus, claims 2-6, depending from claim 1, are not unpatentable over Buse in view of Cheshire and in further view of Reed.

Thus, for at least the reasons set forth above, Appellants submit that claims 2-6 are patentable in their present respective forms.

Accordingly, Appellants respectfully request that the Board reverse the rejection of claims 2-6 under 35 U.S.C. 103(a).

5. CLAIMS 18-22 ARE PATENTABLE OVER BUSE IN VIEW OF CHESHIRE, AND IN FURTHER VIEW OF REED

Claims 18-22 were rejected on the same basis as claims 2-6, and are believed allowable for substantially the same reasons as set forth above with respect to claims 2-6.

Thus, for at least the reasons set forth above, Appellants submit that claims 18-22 are patentable in their present respective forms.

Accordingly, Appellants respectfully request that the Board reverse the rejection of claims 18-22 under 35 U.S.C. 103(a).

C. CLAIMS 7, 11-16, AND 23 ARE PATENTABLE UNDER 35 U.S.C. 103(a)

In the Final Office Action dated October 4, 2005, claims 7, 11-16, and 23 were rejected under 35 U.S.C. §103(a) as being unpatentable over Buse in view of Cheshire, and in further view of Mellquist, U.S. Patent No. 6,115,545 (hereinafter, Mellquist).

However, in determining whether obviousness is established by combining the teachings of the prior art, “the test is what the combined teachings of the references would have suggested to those of ordinary skill in the art,” and the combined teachings of the prior

art references must suggest, expressly or by implication, the improvements embodied by the invention. *In re GPAC Inc.* 35 USPQ2d 1116, 1123 (Fed Cir. 1995).

However, as set forth below, Appellants submit that claims 7, 11-16, and 23 are not disclosed, taught, or suggested by Buse in view of Cheshire, and in further view of Mellquist, and are therefore patentable in their present form.

1. BUSE

Buse is summarized at the beginning of section VIII(A)(1) of this Brief, which for the sake of brevity is not repeated here.

2. CHESHIRE

Cheshire is summarized at the beginning of section VIII(A)(2) of this Brief, which for the sake of brevity is not repeated here.

3. MELLQUIST

Mellquist discloses as background, the use of a BOOTstrap Protocol (BOOTP) that allows clients to automatically receive all IP configuration information from a configured BOOTP server (col. 2, lines 26-30). In order to define an IP address, a free address in the range of valid addresses must be selected (col. 3, lines 12-14). Addresses are usually administered by a person who allocates these addresses to entities who require them (col. 3, lines 14-15). It is important that duplicate addresses are not allowed since this can cause major trouble (col. 3, lines 16-17). Also, a sub-net mask is required for proper operation, and must be the same on all entities across the sub-net (col. 3, lines 17-19).

The Mellquist apparatus includes a configuration module 41 that acts in place of a BOOTP server to accept and reply to a select set of BOOTP requests from devices, wherein the BOOTP response contains an IP address corresponding to a media access control (MAC)

address for the device that submitted the BOOTP request (col. 5, lines 36-45). Once powered up, a network device 33 issues a broadcast BOOTP request 47 which will be picked up by IP configuration module 41, that issues a BOOTP response 48 by which network device 33 will obtain the IP configuration parameters and proceed to initialize (col. 5, line 66 to col. 6, line 5).

**4. CLAIM 7 IS PATENTABLE OVER BUSE IN VIEW OF CHESHIRE, AND
IN FURTHER VIEW OF MELLQUIST**

Claim 7 is directed to the method of claim 1, wherein prior to performing said generating step. Claim 7 recites, broadcasting a discovery packet on said network; receiving a response from said network adapter; and determining if said network adapter has a valid internet protocol address.

As set forth above with respect to claim 1, Buse and Cheshire, taken alone or in combination, would not yield the subject matter of claim 1, and the subject matter of claim 1 is not obvious over Buse in view of Cheshire.

Appellants respectfully submit that Mellquist does not overcome the deficiency of Buse in view of Cheshire, as applied to claim 1, nor does the Examiner assert as much.

Claim 7 is thus believed allowable due to its dependence on otherwise allowable base claim 1.

For example, in rejecting claim 1, the Examiner acknowledges that Buse does not disclose, teach, or suggest the use of an ARP probe “nor where assigning an IP address to a device is performed by assigning the IP address to the network adapter of the device which connects the device to the network (Page 3 of Office Action mailed 10/4/05).

Rather, the Examiner relies upon Cheshire for as disclosing “where assigning an IP address to a device is performed by assigning the IP address to the network adapter of the device which connects the device to the network.”

However, as set forth above with respect to claim 1, Cheshire simply does not make up for the deficiency of the Buse disclosure as applied to claim 1.

In addition, Mellquist does not make up for the deficiency of Buse and Cheshire as applied to claim 1.

Like Cheshire, Mellquist discloses self-configuration of an IP address.

For example, Mellquist et al, discloses that a network device 33 sends out a BOOTP request, and IP configuration module 41, standing in the place of a BOOTP server, provides a BOOTP response including an IP address to network device 33, which then proceeds to initialize. Thus network device 33 configures itself by obtaining an IP address from IP configuration module 41 that acts in the place of a BOOTP server, a process which is known in the art to be self-configuration.

In contrast to a network device that configures itself based on submitting a BOOTP request and receiving a BOOTP response, as disclosed by Mellquist, claim 1 contemplates a computer that performs the step of assigning the internet protocol address to the network device, i.e., a network adapter associated with a device other than the computer that assigns the IP address, via the network.

Unlike the Mellquist disclosure, Appellants’ invention allows the use of a network adapter that is unable to configure itself, i.e., a network adapter that does not contain a mechanism for obtaining an IP address, and depends on another computer to do so (see Appellants’ specification at page 4, lines 28-31).

Thus, since Mellquist does not overcome the deficiency of Buse and Cheshire as applied to claim 1, the combination of Buse, Cheshire, and Mellquist would not yield Appellants' claimed invention.

Although the Examiner asserts in the Response to Arguments that arguments against a reference individually cannot show nonobviousness where the rejections are based on a combination of references, Appellants respectfully submit that, as set forth above, the combination of Buse, Cheshire, and Mellquist would not yield Appellants' claimed invention, since all of the limitations of claim 1 are not taught, disclosed, or suggested by Buse, Cheshire, and Mellquist, taken alone or in combination.

MPEP 2142 provides that to establish a prima facie case of obviousness the prior art reference (or references when combined) must teach or suggest all the claim limitations.

Since Buse, Cheshire, and Mellquist, taken alone or in combination, do not disclose, teach, or suggest all the limitations of claim 1, claim 1 a case of prima facie obviousness has not been established against claim 1, and hence, claim 1 is allowable over Buse in view of Cheshire and in further view of Mellquist.

Claim 7 is thus believed allowable due to its dependence, directly or indirectly, on otherwise allowable base claim 1.

In addition, claim 7 recites, in part, determining if the network adapter has a valid internet protocol address.

In rejecting claim 7, the Examiner relies on Mellquist at column 3, lines 11-19.

Appellants respectfully submit that the relied-upon language of Mellquist merely discloses that a required free address in the range of valid addresses must be selected (col. 3, lines 12-14), that addresses are usually administered by a person who allocates these

addresses to entities who require them (col. 3, lines 14-15), and that duplicate addresses are not allowed (col. 3, lines 16-17).

However, such language simply does not disclose, teach, or suggest any “determination” aspect, much less determining if the network adapter has a valid internet protocol address, as recited in claim 7.

In the Response to Arguments, the Examiner asserts that “The broadest reasonable interpretation has been applied to the claim term “valid internet protocol address.”

However, Appellants respectfully submit that the Patent and Trademark Office determines the scope of claims in patent applications not solely on the basis of the claim language, but upon giving claims their broadest reasonable construction “in light of the specification as it would be interpreted by one of ordinary skill in the art.” *In re Am. Acad. of Sci. Tech. Ctr.*, 367 F.3d 1359, 1364 [70 USPQ2d 1827] (Fed. Cir. 2004) (Emphasis added).

In order to clarify determining if the network adapter has a valid internet protocol address, as recited in claim 7, Appellants respectfully direct the Board’s attention to Appellants’ specification at page 5, lines 25-32, which is reproduced as follows:

At step 104, computer 12 evaluates the response from LCNA 24 to determine if LCNA 24 has a valid IP address. An IP address is considered valid if it is an appropriate address for the subnet to which computer 12 is connected. An uninitialized LCNA always has an invalid IP address. The determination of validity is accomplished by comparing the value associated with the IP address of LCNA 24 to the IP address of computer 12 and a subnet mask of computer 12. If the IP address is valid, then the process terminates at step 120. Otherwise, the process flow continues at step 106.

Mellquist simply does not disclose, teach, or suggest determining if the network adapter has a valid internet protocol address. For example, the relied-upon Mellquist statements merely lists two existing constraints on IP addresses, and disclose that addresses

are administered by a person who allocates the addresses, without stating that there is a determination as to whether an address is valid.

The relied upon Mellquist text simply does not disclose, teach, or suggest finding out if the address is valid by investigation, comparison with known values, reasoning, or calculation, as would constitute determining if the network adapter has a valid internet protocol address as would be interpreted by one of ordinary skill in the art in light of Appellants' specification.

Rather, the relied upon text simply indicates that a free address must be used, indicates who usually provides the addresses, and indicates that duplicate addresses are not allowed, without a determining aspect within the context of Appellants' claimed invention.

Although Buse discloses checking for an IP address conflict, and Cheshire discloses determining whether an address is already in use, neither Buse nor Cheshire disclose, teach, or suggest determining if the network adapter has a valid internet protocol address in the context of Applicants' claimed invention.

Thus, Buse, Cheshire, and Mellquist, taken alone or in combination, do not disclose, teach, or suggest determining if the network adapter has a valid internet protocol address as would be interpreted by one of ordinary skill in the art in light of Appellants' specification, and hence, the combination of Buse, Cheshire, and Mellquist would not yield Appellants' claimed invention.

Accordingly, claim 7 is believed allowable in its own form.

Notwithstanding the above, since Mellquist discloses self-configuration, for substantially the same reasons as set forth above with respect to claim 1 regarding Buse in view of Cheshire, it would not have been obvious to combine the teachings of Cheshire and

Mellquist, i.e., self-configuration, with the allocation of IP addresses by proxy, as disclosed by Buse.

For example, the mere fact that the prior art could be so modified would not have made the modification obvious unless the prior art suggested the desirability of the modification. In re Laskowski 10 USPQ2d 1397 (Fed. Cir. 1989). In addition, prior art references in combination do not make the invention obvious unless something in prior art references would suggest advantage to be derived from combining their teachings. In re Sernaker 217 USPQ 1 (Fed. Cir. 1983).

Also, MPEP 2144 provides that the expectation of some advantage is the strongest rationale for combining references.

However, there is nothing disclosed, taught, or suggested in either of the Buse or Cheshire or Mellquist references that would suggest the desirability of the asserted combination or that there would be an advantage to be derived from their teachings. Thus, it would not have been obvious to combine the prior art references since there is simply nothing in those references suggests that their teachings could be successfully combined to yield advantageous results in the primary reference. In re Sernaker 217 USPQ 1 (Fed. Cir. 1983).

In addition, Appellants submit that since Cheshire and Mellquist are directed to self-configuration, in contrast to Buse, one would not have been motivated to modify Buse, since Buse is directed to configuration by proxy, and also discloses what is purported to be operational method to perform a configuration for a device by proxy that does not purport to need modification in order to achieve its desired result. That is, there would be no motivation to modify a method for configuration by proxy that is operational in of itself

(Buse), much less by incorporating aspects of a method for self-configuration (Cheshire and/or Mellquist).

Appellants further contend that the way in which the Examiner has assembled the combination of Buse, Cheshire and Mellquist, without any advantage disclosed, taught, or suggested by the references, is tantamount to impermissible hindsight reconstruction of Appellants' claims.

It is impermissible to use the claimed invention as an instruction manual or "template" to piece together the teachings of the prior art so that the claimed invention is rendered obvious. In re Fritch, (CA FC) 23 USPQ2d 1780, 1784 (Fed. Cir. 1992). For example, Buse is directed to configuration by proxy, whereas Cheshire and Mellquist are directed to self-configuration.

Since there is no advantage disclosed, taught, or suggested by any of the references (Buse, Cheshire, and/or Mellquist) to modify Buse with Cheshire or Mellquist, it follows that there would not be a motivation to combine the references, and consequently, Appellants submit that the asserted combination is based on impermissible hindsight reconstruction.

Accordingly, Appellants respectfully submit that it would not be obvious to modify Buse with Cheshire and Reed, and thus, claim 7, depending from claim 1, is not unpatentable over Buse in view of Cheshire and in further view of Mellquist.

Thus, for at least the reasons set forth above, Appellants submit that claim 7 is patentable in its present form.

Accordingly, Appellants respectfully request that the Board reverse the rejection of claim 7 under 35 U.S.C. 103(a).

**5. CLAIM 11 IS PATENTABLE OVER BUSE IN VIEW OF CHESHIRE,
AND IN FURTHER VIEW OF MELLQUIST**

Claim 11 is directed to a method of automatically assigning an internet protocol address to a device. Claim 11 recites, in part, determining if said low-cost network adapter has a valid internet protocol address. For substantially the same reasons as set forth above with respect to claim 7, Appellants respectfully submit that Buse in view of Cheshire, and in further view of Mellquist does not disclose, teach, or suggest determining if the low-cost network adapter has a valid internet protocol address.

Claim 11 also recites, in part, the computer performing the steps of: generating an internet protocol address; incorporating said internet protocol address in an address resolution protocol probe; sending said address resolution protocol probe on said network; and determining whether a response to said address resolution protocol probe indicates that said internet protocol address is in use; wherein if said internet protocol address is not in use, then performing the step of assigning said internet protocol address to said low-cost network adapter via said network.

Claim 11 is believed allowable over Buse in view of Cheshire for substantially the same reasons set forth above with respect to claims 1 and 7, since as set forth above with respect to claim 7, Mellquist does not overcome the deficiency of Buse and Cheshire as applied to claim 1, nor does the Examiner assert as much.

Accordingly, for at least the reasons set forth above, Buse in view of Cheshire, and in further view of Mellquist, taken alone or in combination, do not disclose, teach, or suggest the subject matter of claim 11, and the combination of Buse, Cheshire, and Mellquist would not yield Appellants invention of claim 11.

Thus, for at least the reasons set forth above, Appellants submit that claim 11 is patentable in its present form.

Accordingly, Appellants respectfully request that the Board reverse the rejection of claim 11 under 35 U.S.C. 103(a).

6. CLAIMS 12-16 ARE PATENTABLE OVER BUSE IN VIEW OF CHESHIRE, AND IN FURTHER VIEW OF MELLQUIST

Each of claims 12-16 depend directly or indirectly from claim 11. As set forth above, Buse, Cheshire, and Mellquist, taken alone or in combination, would not yield the subject matter of claim 11, and the subject matter of claim 11 is not obvious over Buse in view of Cheshire.

Accordingly, claims 12-16 are believed allowable due to their dependence on otherwise allowable base claim 11.

Thus, for at least the reasons set forth above, Appellants submit that claims 12-16 are patentable in its present form.

Accordingly, Appellants respectfully request that the Board reverse the rejection of claims 12-16 under 35 U.S.C. 103(a).

7. CLAIM 23 IS PATENTABLE OVER BUSE IN VIEW OF CHESHIRE, AND IN FURTHER VIEW OF MELLQUIST

Claim 23 is directed to the system of claim 17. Claim 23 recites, in part, wherein the computer executes preliminary instructions to determine if said network adapter has a valid internet protocol address. For substantially the same reasons as set forth above with respect to claim 7, Buse in view of Cheshire, and in further view of Mellquist do not disclose, teach, or suggest wherein the computer executes preliminary instructions to determine if the

network adapter has a valid internet protocol address, as recited in claim 23. Accordingly, claim 23 is believed allowable in its present form.

In addition, claim 23 is believed allowable due to its dependence on otherwise allowable base claim 17.

Thus, for at least the reasons set forth above, Appellants submit that claim 23 is patentable in its present form.

Accordingly, Appellants respectfully request that the Board reverse the rejection of claim 23 under 35 U.S.C. 103(a).

D. CLAIMS 8 AND 24 ARE PATENTABLE UNDER 35 U.S.C. 103(a)

In the Final Office Action dated October 4, 2005, claims 8 and 24 were rejected under 35 U.S.C. §103(a) as being unpatentable over Buse in view of Cheshire, in further view of Mellquist, and in further view of Troll, Request for Comments: 2563, May 1999, Troll R. (hereinafter, Troll).

However, in determining whether obviousness is established by combining the teachings of the prior art, “the test is what the combined teachings of the references would have suggested to those of ordinary skill in the art,” and the combined teachings of the prior art references must suggest, expressly or by implication, the improvements embodied by the invention. *In re GPAC Inc.* 35 USPQ2d 1116, 1123 (Fed Cir. 1995).

However, as set forth below, Appellants submit that claims 8 and 24 are not disclosed, taught, or suggested by Buse in view of Cheshire, in further view of Mellquist, and in further view of Troll, and are therefore patentable in their present form.

1. BUSE

Buse is summarized at the beginning of section VIII(A)(1) of this Brief, which for the sake of brevity is not repeated here.

2. CHESHIRE

Cheshire is summarized at the beginning of section VIII(A)(2) of this Brief, which for the sake of brevity is not repeated here.

3. MELLQUIST

Mellquist is summarized at the beginning of section VIII(C)(3) of this Brief, which for the sake of brevity is not repeated here.

4. TROLL

Troll is directed to disabling stateless auto-configuration in IPv4 clients (page 1), and allowing a DHCP client to determine whether or not it should assign itself a “link-local” address (page 2). Troll also discloses an auto-configure option which allows a DHCP client to determine whether or not it should generate a link-local IP address.

5. CLAIM 8 IS PATENTABLE OVER BUSE IN VIEW OF CHESHIRE, IN FURTHER VIEW OF MELLQUIST, AND IN FURTHER VIEW OF TROLL

Claim 8 is directed to the method of claim 7, wherein prior to performing said generating step said method comprising the step of determining whether said network allows said computer to assign an internet protocol address to said network adapter.

Claim 8 depends from claim 7, which depends from claim 1. As set forth above with respect to claim 7, the subject matter of either of claims 1 and 7 is not unpatentable over Buse in view of Cheshire, and in further view of Mellquist. Appellants respectfully submit

that Troll does not overcome the deficiency of Buse in view of Cheshire, and in further view of Mellquist, nor does the Examiner assert as much.

For example, as set forth above, and as acknowledged by the Examiner, Troll is directed to a DHCP client assigning itself an IP address, which is known in the art as self assignment, or self configuration.

In contrast, however, claims 1 and 7 contemplate a computer that performs the step of assigning the internet protocol address to the network device, i.e., the network adapter, via the network, which allows the use of a network adapter that is unable to configure itself, i.e., a network adapter that does not contain a mechanism for obtaining an IP address, and depends on another computer to do so (see Appellants' specification at page 4, lines 28-31).

As set forth above with respect to claims 1 and 7, Buse in view of Cheshire and in further view of Mellquist would not yield Appellants' invention of claim 7, and that it would not be obvious to combine the teaching of Buse in view of Cheshire.

For substantially the same reasons as set forth above with respect to claims 1 and 7 as with respect to Buse in view of Cheshire, it would not have been obvious to combine the teachings of Cheshire, Mellquist, and Troll, i.e., self-configuration, with the allocation of IP addresses by proxy, as disclosed by Buse.

Accordingly, claim 8 is not unpatentable over Buse in view of Cheshire, in further view of Mellquist, and in further view of Troll, due to its dependence on otherwise allowable base claim 1 and intervening claim 7.

In addition, Troll does not disclose, teach, or suggest determining whether the network allows the computer to assign an internet protocol address to the network adapter, as recited in claim 8, nor do the other cited references.

Rather, Troll discloses that a DHCP client will be able to determine whether the network is centrally administered, thus allowing it to determine whether or not it should assign itself an address (page 2).

Troll also discloses that a DHCP client will be allowed to determine whether or not it should generate an address (page 3). However, the relied-upon Troll disclosures have no bearing on and do not disclose, teach, or suggest determining whether the network allows the computer to assign an internet protocol address to the network adapter, as recited in claim 8.

In the Response to Arguments, the Examiner asserts that “The broadest reasonable interpretation has been applied to the claims, and that “a network that allows a computer to assign an internet protocol address to a network adapter” “simply means determining that DHCP server(s) are available on the network from which remote automatic IP addresses can be obtained for providing to a devices’ network adapter.”

However, Appellants respectfully submit that the Patent and Trademark Office determines the scope of claims in patent applications not solely on the basis of the claim language, but upon giving claims their broadest reasonable construction “in light of the specification as it would be interpreted by one of ordinary skill in the art.” *In re Am. Acad. of Sci. Tech. Ctr.*, 367 F.3d 1359, 1364 [70 USPQ2d 1827] (Fed. Cir. 2004) (Emphasis added).

In order to clarify determining whether the network allows the computer to assign an internet protocol address to the network adapter, as recited in claim 8, Appellants respectfully direct the Board’s attention to Appellants’ specification at page 5, line 33 to page 6, line 9 which is reproduced as follows:

At step 106, computer 12 determines if network 16 allows automatic remote assignment of IP addresses. If network 16 allows automatic remote assignment of IP addresses, then process flow continues at step 108. Otherwise, the process terminates at step 120. Computer 12 provides for the

manual assignment of an IP address, which is not a part of this invention, thus in the event network 16 does not allow automatic remote assignment of IP addresses, an IP address can be assigned manually.

Determination as to whether network 16 allows the assignment of IP addresses to LCNA type devices is necessary since some network environments do not allow for automatic remote IP address assignment. If the network environment utilizes certain addresses, such as those used by the UPnP or APIPA addressing schemes, then automatic remote IP address assignment is possible.

Appellants further respectfully direct the Board's attention to Appellants' specification at page 1, line 29 to page 2, line 14 which is reproduced as follows:

There are several industry standards by which a network device can automatically obtain an IP address information. Such standards include the aforementioned DHCP, Universal Plug and Play (UPnP) and other forms of Automatic Private IP Addressing (APIPA). Each of these standards require that significant network transactions be initiated and conducted by the network device itself which requires hardware and configuration storage, making them cost prohibitive for low-cost devices.

What is needed in the art is an apparatus and a method by which a device on a computer network can be assigned an IP address automatically, without the overhead of supporting traditional address assignment protocols.

The present invention provides an apparatus and a method by which a device on a computer network can be assigned an IP address automatically, without the overhead of supporting the traditional address assignment protocols, such as DHCP, within the devices themselves. (Emphasis added).

Appellants respectfully submit that upon giving claim 8 its broadest reasonable construction "in light of the specification as it would be interpreted by one of ordinary skill in the art," one of ordinary skill in the art would not interpret "a network that allows a computer to assign an internet protocol address to a network adapter" to "simply mean[s] determining that DHCP server(s) are available on the network from which remote automatic IP addresses can be obtained for providing to a devices' network adapter," as asserted by the examiner.

For example, Appellant's specification is clearly directed to a method by which a device on a computer network can be assigned an IP address automatically, without the overhead of supporting the traditional address assignment protocols, such as DHCP, within the devices themselves (Spec at page 2, lines 11-14).

Appellants respectfully submit that the fact of the Examiner's reliance on a combination of 4 references, in the manner set forth to reject claim 8, supports Appellants' present contention of impermissible hindsight reconstruction of Appellants invention, using Appellants' claims as a blueprint, for at least the reasons set forth above with respect to claim 1.

Thus, for at least the reasons set forth above, Appellants submit that claim 8 is patentable in its present form.

Accordingly, Appellants respectfully request that the Board reverse the rejection of claim 8 under 35 U.S.C. 103(a).

6. CLAIM 24 IS PATENTABLE OVER BUSE IN VIEW OF CHESHIRE, IN FURTHER VIEW OF MELLQUIST, AND IN FURTHER VIEW OF TROLL

Claim 24 was rejected on the same basis as claim 8, and is believed allowable for substantially the same reasons as set forth above with respect to claims 8.

Thus, for at least the reasons set forth above, Appellants submit that claim 24 is patentable in its present form.

Accordingly, Appellants respectfully request that the Board reverse the rejection of claim 24 under 35 U.S.C. 103(a).

E. CONCLUSION

For the foregoing reasons, Appellants submit that claims 1-25 are patentable in their present respective forms. Accordingly, Appellants respectfully requests that the Board reverse the final rejections of the appealed claims.

Respectfully submitted,



Paul C. Gosnell
Registration No. 46,735

Attorney for Appellants

PCG14/ts

TAYLOR & AUST, P.C.
12029 E. Washington Street
Indianapolis, IN 46229
Telephone: 317-894-0801
Facsimile: 317-894-0803

Encs.: Return postcard

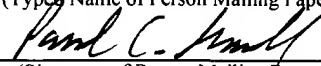
"EXPRESS MAIL" Mailing Number EV 620802603 US

Date of Deposit March 3, 2006

I hereby certify that this paper or fee is being deposited with the United States Postal Service "EXPRESS MAIL POST OFFICE TO ADDRESSEE" service under 37 CFR 1.10 on the date indicated above and is addressed to the MS APPEAL BRIEF - PATENTS Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450

Paul C. Gosnell, Reg. No. 46,735

(Typed Name of Person Mailing Paper or Fee)



(Signature of Person Mailing Paper or Fee)

IX. CLAIMS APPENDIX

1. A method of automatically assigning an internet protocol address to a device,
comprising the steps of:
 - providing a network;
 - providing a computer communicatively coupled to said network;
 - 5 providing a network adapter to communicatively couple said device to said network,
said network providing communicative interconnection between said computer and said
network adapter;
 - said computer performing the steps of:
 - generating an internet protocol address;
 - 10 incorporating said internet protocol address in an address resolution
protocol probe;
 - sending said address resolution protocol probe on said network; and
 - determining whether a response to said address resolution protocol probe indicates
that said internet protocol address is in use;
 - 15 wherein if said internet protocol address is not in use, then performing the step of
assigning said internet protocol address to said network adapter via said network.
2. The method of claim 1, wherein if said internet protocol address is in use, then
further comprising the step of repeating said generating step, said incorporating step, said
sending step and said determining step.
3. The method of claim 2, further comprising the step of counting a number of
times said generating step is performed.
4. The method of claim 3, comprising the step of comparing said number of times
said generating step is performed to a predetermined number.
5. The method of claim 4, wherein said predetermined number is at least 30.

6. The method of claim 4, wherein if said number of times said generating step is performed exceeds said predetermined number then said computer does not automatically assign said network adapter an internet protocol address.

7. The method of claim 1, wherein prior to performing said generating step, said method comprising the steps of:

broadcasting a discovery packet on said network;

receiving a response from said network adapter; and

5 determining if said network adapter has a valid internet protocol address.

8. The method of claim 7, wherein prior to performing said generating step said method comprising the step of determining whether said network allows said computer to assign an internet protocol address to said network adapter.

9. The method of claim 1, wherein said device is a printer.

10. The method of claim 1, wherein said network adapter is a low-cost network adapter.

11. A method of automatically assigning an internet protocol address to a device, comprising the steps of:

providing a network;

providing a computer communicatively coupled to said network;

5 providing a low-cost network adapter to communicatively couple said device to said network, said network providing communicative interconnection between said computer and said low-cost network adapter;

said computer performing the steps of:

broadcasting a discovery packet on said network;

10 receiving a response from said low-cost network adapter;

determining if said low-cost network adapter has a valid internet protocol address;

wherein if said low-cost network adapter does not have a valid internet protocol address, then said computer performing the steps of:

15 generating an internet protocol address;
 incorporating said internet protocol address in an address resolution protocol probe;
sending said address resolution protocol probe on said network; and

20 determining whether a response to said address resolution protocol probe indicates that said internet protocol address is in use;

 wherein if said internet protocol address is not in use, then performing the step of assigning said internet protocol address to said low-cost network adapter via said network.

12. The method of claim 11, wherein if said internet protocol address is in use, then further comprising the step of repeating said generating step, said incorporating step, said sending step and said determining step.

13. The method of claim 12, further comprising the step of counting a number of times said generating step is performed.

14. The method of claim 13, comprising the step of comparing said number of times said generating step is performed to a predetermined number.

15. The method of claim 14, wherein said predetermined number is at least 30.

16. The method of claim 14, wherein if said number of times said generating step is performed exceeds said predetermined number then said computer does not automatically assign said low-cost network adapter an internet protocol address.

17. A network based imaging system, comprising:

a network;

a computer communicatively coupled to said network;

an imaging device; and

5 a network adapter communicatively coupling said imaging device to said network, said network providing communicative interconnection between said computer and said network adapter;

wherein said computer executes instructions which generate an internet protocol address, incorporate said internet protocol address into an address resolution protocol probe, send said address resolution protocol probe on said network, utilize a response to said address resolution protocol probe to determine if said internet protocol address is in use and if said internet protocol address is not in use, then assign said internet protocol address to said network adapter via said network.

10

18. The system of claim 17, wherein if said internet protocol address is in use then said computer repeats said instructions.

19. The system of claim 18, wherein said computer counts a number of times said instructions are executed.

20. The system of claim 19, wherein said computer compares said number of times said instructions are executed to a predetermined number.

21. The system of claim 20, wherein said predetermined number is at least 30.

22. The system of claim 20, wherein if said number of times said instructions are executed exceeds said predetermined number then said computer does not automatically assign said network adapter an internet protocol address.

23. The system of claim 17, wherein prior to performing said instructions said computer executes preliminary instructions which broadcast a discovery packet on said

network, receive a response from said network adapter and determine if said network adapter has a valid internet protocol address.

24. The system of claim 23, wherein said preliminary instructions further determine whether said network allows said computer to assign an internet protocol address to said network adapter.

25. The system of claim 17, wherein said network adapter is a low-cost network adapter.

X. EVIDENCE APPENDIX

Included herein, and listed below, is a copy of each reference upon which the Examiner relied in rejecting one or more of the claims of the present application.

Exhibit:

- A. U.S. Patent No. 6,810,420 B1 (Buse).
- B. Cheshire, S., Current Meeting Report, Cheshire, et al., 03/99 (Cheshire)
- C. U.S. Patent No. 6,061,739 (Reed).
- D. U.S. Patent No. 6,115,545 (Mellquist).
- E. Troll, Request for Comments: 2563, May 1999, Troll R. (Troll)

XI. RELATED PROCEEDINGS APPENDIX

(No Entries)

- Current Meeting Report
- Slides

2.3.10 Networks in the Small - aka Home Networks (nits) bof

Current Meeting Report

Minutes: NITS BOF

15 March, 1999

Minneapolis IETF

CoChairs:

Stuart Cheshire, cheshire@apple.com

Peter Ford, peterf@microsoft.com

Mailing list: nits@merit.edu

After a short welcome and discussion of agenda we established a short list of presentations to discuss home networks. It appears that about 180-200 people attended the BOF, with a full room.

The list of presentations included:

- 1) Overall Goals of NITS BOF - Peter Ford
- 2) Home Networking Device and Service Discovery Requirements
- draft-miller-homedisc-req-00.txt - Brent Miller
- 3) Automatic IP address assignment for link local address w/IPv4
- draft-ietf-dhc-ipv4-autoconfig-03.txt - Stuart Cheshire/Ryan Troll
- 4) Multicast Discovery of DNS Services
- draft-manning-multicast-dns-01.txt - Bill Woodcock/Bill Manning
- 5) Service Location Protocol
- draft-ietf-srvloc-protocol-v2-12.txt - Erik Guttman
- 6) Simple Service Discovery Protocol
- draft-cai-ssdp-v1-00.txt - Yaron Goland
- 7) IPv6 Autoconfiguration and Neighbor Discovery - T. Narten
- 8) IP Address sharing - Stuart Cheshire

Goals of NITS BOF

Presented by Peter Ford and Stuart Cheshire

Goals articulated by Peter:

- 1) Discuss How Networks in the Small (e.g. Home Networks) can and should work
- 2) Determine if there is standards work to be done
- 3) If so, charter appropriate work
- 4) Non-goal - solve all problems by 5.30pm

Why should the IETF consider working on NITS now?

- 1) IP stack has sufficiently standardized and many vendors are looking at IP based devices that should have "plug and play behaviour".
- 2) IP currently fails the simplicity test - if one were to go to a CostCO and buy 2 PCs and an ethernet hub someone would still have to configure the IP addressing and other stack parameters.

Peter suggested that we should benchmark current IPv4 practices against those of Appletalk, IPX, and NetBIOS, including deployment of DHCP servers and bootstrapping thru DHCP to get DNS server addresses and domains configured in an end system. When one considers that to print many people then have to go edit their /etc/printcap and /etc/resolv.conf files we should turn red in embarrassment. With Appletalk and NetBIOS over NetBeui or IPX you can bring a system up and print with minimal (close to zero) configuration.

Beyond simple addressing and DNS configuration we also need to worry about router, proxy and NAT configs that get more daunting every day.

Brent Miller from IBM Raleigh went through a brief description of the requirements his team from IBM developed for Home networks.

Basic assumptions include that there is a home LAN that is intermittently connected to the Internet and the LAN and LAN clients use standard IETF protocols. A computer that is introduced to this LAN should be able to enjoy basic connectivity to other similar systems and to services on that network. IN short, things should "just work".

Brent discussed a taxonomy of requirements from the draft including:

Autoconfiguration: IP address assignment to new devices, Service/Device location, and the use of User friendly names.

Question: Is this limited to single subnet single domain? What about multiple administrative domains?

What if your teenage children want to run their own autonomous domains at home?

Answer: In order to make progress, should not require NITS solutions to scale beyond single subnet, single administrative domain.

Automatic IP address assignment for link local address w/IPv4

- draft-ietf-dhc-ipv4-autoconfig-03.txt - Stuart Cheshire/Ryan Troll

Stuart Cheshire presented a basic overview of IPv4 address self configuration as currently implemented by Apple and MS. The purpose is to support configuration of basic stacks without manual configuration. Stuart pointed out that both Apple and Microsoft are attempting to move towards complete use of IETF standard protocols in replacement of their legacy protocols (Appletalk and NetBeui).

Stuart started off his slide deck with a simple description of autonet:

- 1) Designed for small isolated LANs
 - a home LAN
 - a small office
- 2) Currently in Windows 98 and MAC OS 8.5
- 3) Described in an Internet Draft by Ryan Troll (Carnegie Mellon).

Stuart proceeded to describe the operation in Mac OS 8.5.

- 1) DHCP Discover
- 2) If no reply retry, 4, 8, 16 second intervals
- 4) If no DHCP server discovered then:
 - 4a) Pick a random address on 169.254.*.* subnet (except first and last 256 addresses which are reserved for future use).

Send an ARP probe (ala DHCP conflict detection) to verify address is not already in use

If address in use, go back to 4a) - iterate at most 10 times and then fail stack initialization

Configure the interface with IP address, and start using interface

Every 5 minutes send single DHCP Discover to determine if a DHCP server has come online on the LAN, if so then proceed to normal DHCP client behavior upon DHCP Offer message reception.

Stuart then pointed out key points/features:

- 1) allows invisible use of IP - a user can go to the chooser using appletalk, but actually connects to it using autoconfigured TCP/IP
- 2) This is NOT a substitute with IPv6 - IPv6 is critical so that everyone can get globally unique public IP addresses
- 3) Stuart would like to also perform resource and service discovery using IP instead of Appletalk Name Binding Protocol (NBP).

There were questions about some of the DHCP specifics. Also several questions came up on who "owned" the 169.254/16 subnet, and Bill Manning offered that the IANA did at that is what WHOIS tells people who see this happening behind their firewalls!

MUDDS

Multicast Discovery of DNS Services

Presented by Bill Woodcock

Bill indicated that version 1 of the spec is in the Internet Draft directory but that he and Bill Manning are up to version 3 of the spec. He noted that people are looking for simple replacements for existing protocols such as Appletalk NBP and NetBIOS browsing.

There are several applications:

- 1) Bootstrapping DNS resolvers on clients when there is not a DHCP server
- 2) Discovery of unconfigured devices such as printers and routers
- 3) Let Apple deprecate the AppleTalk Chooser

Bill also noted that this mechanism could be used to provide a lightweight subset of SLP features. Steve Deering asked how this works and Bill briefly described how you can find SRV records on hosts that are listening to the multicast port. A question was asked about the hostname part of the SRV record if you were looking for "any printer on the LAN" and Bill noted that DNS supports the use of wildcards (*) in the names, something which raised several eyebrows in the audience. Erik Guttman asked about how do you get DNS response suppression by end nodes if a DNS server WAS present on a LAN to which Bill described how one would first look for a DNS server via multicast and if a server was found you would only use unicast queries to that server. This raised several questions about the scoping of multicast requests and several references to the work in the multicast working groups on administratively scoped multicast.

Erik Guttman noted that you need to put in aggressive backoffs into the protocol to prevent swamping of the LAN, especially in wireless cases. He also noted that there were issues of trust models, which are contained in SLP.

Henry Sinnreich of MCI asked several questions about whether this mechanism could be extended to find public DNS servers. This was answered that it depends on careful configuration of multicast scopes.

Bill pointed out that the major advantage of using multicast for DNS discovery is that it uses pre-existing technologies: Multicast and DNS. The multicast usage employs a statically assigned address within the administrative local-scope and SRV records are used to describe network services such as DNS or printing. The DNS transaction works as normal except the initial query is performed within the multicast group rather than with a preconfigured DNS server.

SLP Overview by Erik Guttman

Erik gave a brief overview of how SLP works and the current standards status. he mentioned several small last minute additions to cover the cases where people are using directories such as LDAP and what a minimal subset of SLP would be. He also described some future work in the SLP group that would address issues that some parties have had with the use of SLP in environments with LDAP servers:

- 1) an LDAP server could emit DA Adverts allowing LDAP clients to use LDAP directly instead of using SLP front ending a DS.
- 2) Erik described how JINI and non-JINI services are discovered using SLP. Erik then compared SLP with multicast DNS:

Queries in SLP are of the form: "servers that ..." such as what are the printers that all have 721 dpi support.

SLP is carefully crafted to work in networks from small to large

Steve Deering proceeded to ask if any consideration was made in SLP to take advantage of many multicast addresses so that the multicast IP filter could filter out many questions irrelevant to the server instead of waking all servers with all SLP queries. Erik said this was in the early versions of the protocol, but that it was pulled out of SLPv2 due to no apparent interest.

Charlie Perkins noted that SLP was designed against a set of requirements that looked surprisingly similar to the requirements that Brent Miller presented at the beginning of the session.

Simple Service Discovery presented by Yaron Goland

Yaron discussed Microsoft's investigations into the subset of home networking including service discovery.

The problem statement is how systems can discover each other even if there is no network administrator or directory support. A subset of this problem is to discover directory support if present on the network.

The target environment anticipated is:

- 1) an IP network with multicast support
- 2) limited memory and storage - aggregate of 1M of memory or less
- 3) HTTP and XML support expected to be common for supporting device to device communication. Yaron noted that investigation of embedded web systems web sites indicate that web servers can be small (< 70Kbytes of code).
- 4) Example devices include: thermostats, security systems, CD players, printers, scanners, etc.

Solution Parameters include:

- 1) multicast IP is used to enable discovery in the absence of directory.
- 2) HTTP/XML based - to re-use stacks already present in the devices (small web servers that support UI, mgmt, etc.)
- 3) no parameters - services are identified with URIs, any more powerful discovery or parameter negotiation is done in a "higher" protocol or a service specific protocol (e.g. IPP).

Yaron noted the draft on SSDP in the Interdraft directory and also noted there were several items TBD. A proposed implementation is:

- 1) use HTTP over multicast UDP
- 2) Declaration based discovery using OPTIONS method and If header
- 3) announcement based discovery using ANNOUNCE method and resource-state header. This would allow new services to introduce themselves on the fly.

There was good followup discussion on the main differences btw SSDP and SLP. SSDP does not support queries of the form "who are the XXX servers that are ...", only dealing with "who are XXX servers?"

IPv6 autoconfiguration and link local addressing - Tom Narten

The goal is to present an overview of what was done and standardized in IPv6 to support small networks

For autoconfiguration IPv6 provides for:

PNP out of the box experience including support for the isolated dentist's office (single LAN that is not Internet connected) with support that scales to large enterprise.

All addresses are "leased"

IPv6 has built in support for multiple addresses per interface.

IPv6 also has scoped addressing:

- 1) Link-Local for communication with immediate neighbor
- 2) Site-Local which is analogous to IPv4 net 10/8
- 3) global addresses that are globally unique

These are all documented in RFCs (question from audience).

Link Local addresses are such that every node assigns link local addresses to its interfaces, those addresses are only good for that link/subnetwork. Link local addresses have a well known prefix (fe80::/9). Each address has an interface identifier that is derived from the MAC address, which is globally unique. There is built in duplicate address detection. It works much like the IPv4 autoconfiguration with a much lower probability of collision - probably none.

IPv6 supports both DHCP and stateless address autoconfiguration where there is no server on the wire (usually the router). A router sends a router advertisement and the end system cons's up an address from a prefix contained in the router advertisement (RA) and the Interface ID derived from the MAC of the Interface. If the interfaces sees multiple RAs then it means the host can have multiple addresses. One changes an interfaces address by changing the prefix contained in the RA. Tom noted that IPv6 uses dynamic DNS to update the DNS when an interface's address changes.

Router advertisements contain a default router list. Each host picks this up and maintains a local list of default routers. The host keeps track of reachability to all neighbors, if necessary the host can send probes.

IP address sharing

Stuart Cheshire/Apple Computer, Inc.

Stuart began describing the current status of getting PacBell ADSL at up to T-1 data rates. To get a single IP address for a single host the service is \$50/month. If you want more IP hosts then it costs >\$100 per month.

Stuart infers that IPv4 addresses are becoming scarce.

People are finding an arbitrage around additional cost per address in the deployment of NATs. He notes this is not an end-all solution. Nats are: fragile, break the end to end model, breaks IPSEC, requires separate support on a per protocol basis such as ftp and any other protocol that buries ports inside PDUs. The result is that NATs hold state on a per connection basis and is a single point of failure.

How could you make hosts behind NATs work better if they did know they were behind a NAT? Stuart poses that there can be address sharing aware (AAA) hosts. You need to ensure that on a single LAN where you are sharing the same IP address across hosts that hosts use unique to the LAN ports. The

NAT would need to be able to demultiplex inbound packets to hosts based on which hosts on the LAN have what ports in use. To that end Stuart proposed an "extended ARP" that is based on <IP address, proto, port> instead of simply using the IP addr. This can be used as a claim mechanism as well as used by the NAT box to deliver a packet to the right host.

Stuart wanted to make it clear this was a 1-2 year solution and that IPv6 should be the real answer in the long term when it is fully deployed.

Several members in the group noted similar ideas were being discussed in the NAT working group and perhaps this topic should be out of scope of NITS due to active work in the NAT WG.

Closing Discussion

There appears to be sufficient interest in the area of small networks to proceed with forming a work list. Stuart and Peter took the action to work out a charter with the ADs (Thomas Narten and Erik Nordmark).

In enumerating the topics to be discussed we collected:

- 1) multicast DNS: more spec work needed, name resolution for IPv4 and v6, service discovery. Many voiced concern over m'cast scoping of queries. This topic had a lg amount of support for wg activity.
- 2) service discovery? mcast DNS for this purpose vs. SLP or SSDP? Issues to be determined was the spectrum of scalability, would mcast DNS suffice? Some mentioned applicability as an issue to be resolved.
- 3) What about IPv6, perhaps we should skip v4 and move directly to v6.
- 4) Need to refine requirements doc to aid in resolution of item 2) above.
- 5) What about multiple LANs (e.g. wireless, 1394 and ethernet all in one house)? Most felt we needed to worry about this topic.
- 6) what about multiple administrative domains on the same LAN such as in apartment buildings?
- 7) security was absent for most of the presentations, what are the security models and there was some who shared that they believed that security in the home was VERY important.
- 8) Is mobility an issue?
- 9) what about intranet vs Internet as a connectivity model (some said the use of the word intranet should be grounds for banishment ...). This appeared in the context of should the addresses on Home LANs be private or public, and how should 2 homes interconnect if private addresses are used

The meeting drew to a close at 6pm.

Slides

None received.

DHCP Option to Disable Stateless Auto-Configuration in IPv4 Clients

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

Abstract

Operating Systems are now attempting to support ad-hoc networks of two or more systems, while keeping user configuration at a minimum. To accommodate this, in the absence of a central configuration mechanism (DHCP), some OS's are automatically choosing a link-local IP address which will allow them to communicate only with other hosts on the same link. This address will not allow the OS to communicate with anything beyond a router. However, some sites depend on the fact that a host with no DHCP response will have no IP address. This document describes a mechanism by which DHCP servers are able to tell clients that they do not have an IP address to offer, and that the client should not generate an IP address it's own.

1. Introduction

With computers becoming a larger part of everyday life, operating systems must be able to support a larger range of operating environments. One aspect of this support is the selection of an IP address. The Dynamic Host Configuration Protocol [DHCP] provides a superb method by which site administrators may supply IP addresses (and other network parameters) to network devices. However, some operating environments are not centrally maintained, and operating systems must now be able to handle this quickly and easily.

IPv6 accounts for this, and allows an IPv6 stack to assign itself a global address in the absence of any other mechanism for configuration [IPv6SAC]. However, Operating System designers can't wait for IPv6 support everywhere. They need to be able to assume

they will have IPv4 addresses, so that they may communicate with one another even in the smallest networks.

This document looks at three types of network nodes, and how IPv4 address auto-configuration may be disabled on a per-subnet (or even per-node) basis. The three types of network nodes are:

- * A node for which the site administrator will hand out configuration information,
- * A node on a network segment for which there is no site administrator, and
- * A node on a network segment that has a central site administrator, and that administrator chooses not to hand out any configuration information to the node.

The difference between the second and third cases is the clients behavior.

In one case, the node may assign itself an IP address, and have full connectivity with other nodes on the local wire. In the last case, the node is not told what to do, and while it may assign itself a network address in the same way as case #2, this may not be what the central administrator wants.

The first scenario is handled by the current DHCP standard. However, the current DHCP specification [DHCP] says servers must silently ignore requests from hosts they do not know. Because of this, DHCP clients are unable to determine whether they are on a subnet with no administration, or with administration that is choosing not to hand out addresses.

This document describes a method by which DHCP clients will be able to determine whether or not the network is being centrally administrated, allowing it to intelligently determine whether or not it should assign itself a "link-local" address.

1.1. Conventions Used in the Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [KEYWORDS].

1.2. Terminology

- DHCP client A DHCP client is an Internet host using DHCP to obtain configuration parameters such as a network address.
- DHCP server A DHCP server is an Internet host that returns configuration parameters to DHCP clients.

2. The Auto-Configure Option

This option code is used to ask whether, and be notified if, auto-configuration should be disabled on the local subnet. The auto-configure option is an 8-bit number.

Code	Len	Value
116	1	a

The code for this option is 116, and its length is 1.

This code, along with the IP address assignment, will allow a DHCP client to determine whether or not it should generate a link-local IP address.

2.1. Auto-Configure Values

The auto-configure option uses the following values:

DoNotAutoConfigure	0
AutoConfigure	1

When a server responds with the value "AutoConfigure", the client MAY generate a link-local IP address if appropriate. However, if the server responds with "DoNotAutoConfigure", the client MUST NOT generate a link-local IP address, possibly leaving it with no IP address.

2.2. DHCP Client Behavior

Clients that have auto-configuration capabilities MUST add the Auto-Configure option to the list of options included in its initial DHCPDISCOVER message. ([DHCP] Section 4.4.1) At this time, the option's value should be set to "AutoConfigure".

When a DHCPOFFER is received, it is handled as described in [DHCP], section 4.4.1, with one exception. If the 'yiaddr' field is 0x00000000, the Auto-Configure option must be consulted. If this

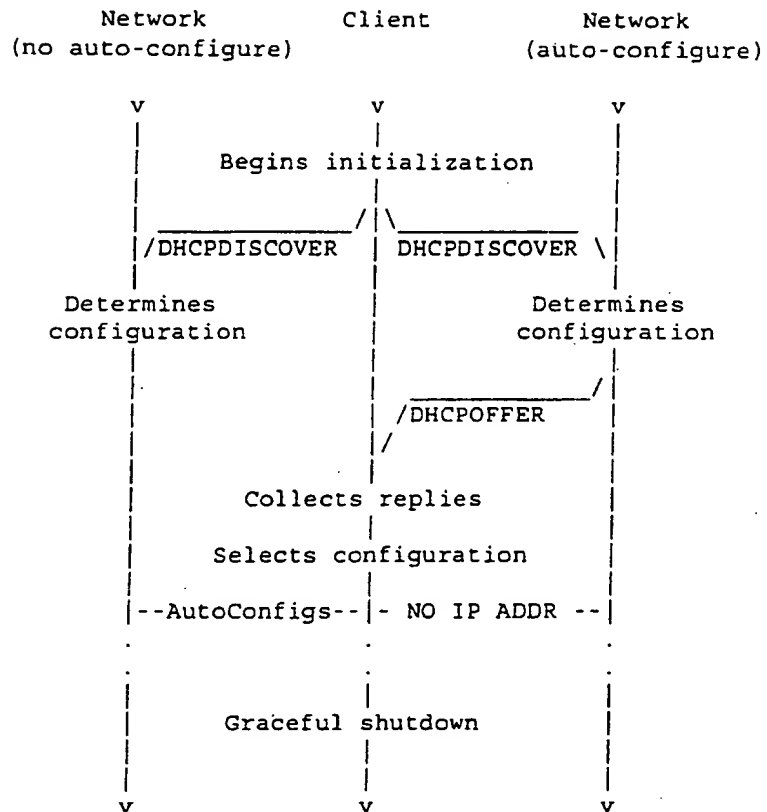
option is set to "AutoConfigure", then the DHCPPOFFER MUST be ignored, and the DHCP client MAY generate a link-local IP address. However, if this option is set to "DoNotAutoConfigure", then the DHCPPOFFER MUST be ignored, and the client MUST NOT generate a link-local IP address.

If a DHCP client receives any DHCPPOFFER which contains a 'yiaddr' of 0x00000000, and the Auto-Configure flag says "DoNotAutoConfigure", in the absence of a DHCPPOFFER with a valid 'yiaddr', the DHCP client MUST NOT generate a link-local IP address. The amount of time a DHCP client waits to collect any other DHCPPOFFERS is implementation dependant.

DHCPPOFFERS with a 'yiaddr' of 0x00000000 will only be sent by DHCP servers supporting the Auto-Configure option when the DHCPDISCOVER contained the Auto-Configure option. Since the DHCPDISCOVER will only contain the Auto-Configure option when a DHCP client knows how to handle it, there will be no inter-operability problems.

If the DHCP server does have an address to offer, the message states are the same as those described in [DHCP], section 3.

The following depicts the difference in responses for non-registered DHCP clients that support the "Auto-Configure" option on networks that have DHCP servers that support auto-configuration and networks with DHCP servers that do not.



2.3. DHCP Server Behavior

When a DHCP server receives a DHCPDISCOVER, it MUST be processed as described in [DHCP], section 4.3.1. However, if no address is chosen for the host, a few additional steps MUST be taken.

If the DHCPDISCOVER does not contain the Auto-Configure option, it is not answered.

If the DHCPDISCOVER contains the Auto-Configure option, and the site administrator has specified that Auto-Configuration should be disabled on the subnet the DHCPDISCOVER is originating from, or for the client originating the request, then a DHCPOFFER MUST be sent to the DHCP client. This offer MUST be for the address 0x00000000, and the Auto-Configure option MUST be set to "DoNotAutoConfigure".

If the site administrator allows auto-configuration on the originating subnet, the DHCPDISCOVER is not answered as before.

2.4. Mixed Environments

Environments containing a mixture of clients and servers that do and do not support the Auto-Configure option will not be a problem. Every DHCP transaction is between a Server and a Client, and the possible mixed scenarios between these two are listed below.

2.4.1. Client Supports, Server Does Not

If a DHCP client sends a request that contains the Auto-Configure tag, a DHCP server that does not know what this tag is will respond normally. According to [DHCP] Section 4.3.1, the server MUST NOT return a value for that parameter.

In this case, the server will either respond with a valid DHCP OFFER, or it will not respond at all. In both cases, a DHCP client that supports this option will never care what the state of the option is, and may auto-configure.

2.4.2. Servers Supports, Client Does Not

If the Auto-Configure option is not present in the DHCPDISCOVER, the server will do nothing about it. The client will auto-configure if it doesn't receive a response and believes that's what it should do.

This scenario SHOULD not occur, as any stacks that implement an auto-configuration mechanism MUST implement this option as well.

2.5. Interaction With Other DHCP Messages

As this option only affects the initial IP address selection, it does not apply to subsequent DHCP messages. If the DHCP client received a lease from a DHCP server, future DHCP messages (RENEW, INFORM, ACK, etc.) have no need to fall over into an auto-configuration state.

If the DHCP client's lease expires, the client falls back into the INIT state, and the initial DHCPDISCOVER is sent as before.

2.5.1. DHCPRELEASE Messages

DHCPRELEASEs occur exactly as described in [DHCP], section 4.4.6. When a DHCP client is done with a lease, it MAY notify the server that it is finished. For this to occur, the DHCP client already received a DHCP lease, and the state of Auto-Configuration on the local wire does not matter.

2.5.2. DHCPDECLINE Messages

A DHCPDECLINE is sent by the DHCP client when it determines the network address it is attempting to use is already in use. As a network address has been tested, it must have been offered by the DHCP Server, and the state of Auto-Configuration on the local wire does not matter.

2.5.3. DHCPINFORM Messages

DHCPINFORMs should be handled as described in [DHCP], section 4.4.3. No changes are necessary.

2.6. Message Option

If the DHCP server would like to tell a client why it is not allowed to auto-configure, it MAY add the Message option to the response. This option is defined in [DHCOPT], Section 9.9.

If the DHCP client receives a response with the Message option set, it MUST provide this information to the administrator of the DHCP client. How this information is provided is implementation dependant.

3. Security Considerations

DHCP per se currently provides no authentication or security mechanisms. Potential exposures to attack are discussed in section 7 of the DHCP protocol specification [DHCP].

This mechanism does add one other potential attack. Malicious users on a subnet may respond to all DHCP requests with responses telling DHCP clients that they should NOT auto-configure on the local wire. On a network where Auto-Configuration is required, this will cause all DHCP clients to not choose an address.

4. Acknowledgments

This idea started at a joint Common Solutions Group / Microsoft meeting at Microsoft in May, 1998. The IP stacks in Win98 and NT5 assign themselves an IP address (in a specific subnet) in the absence of a responding DHCP server, and this is causing headaches for many sites that actually rely on machines not getting IP addresses when the DHCP servers do not know them.

Walter Wong proposed a solution that would allow the DHCP servers to tell clients not to do this. His initial solution would not work without slight modifications to DHCP itself. This document describes

those modifications.

5. IANA Considerations

The IANA has assigned option number 116 for this option.

6. References

- [DHCP] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.
- [DHCP OPT] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extension", RFC 2132, March 1997.
- [IPv6SAC] Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 2462, December 1998.
- [KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

7. Author's Address

Ryan Troll
@Home Network
425 Broadway
Redwood City, CA 94063

Phone: (650) 556-6031
EMail: rtroll@corp.home.net

8. Full Copyright Statement

Copyright (C) The Internet Society (1999). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

An Ethernet Address Resolution Protocol
-- or --
Converting Network Protocol Addresses
to 48.bit Ethernet Address
for Transmission on
Ethernet Hardware

Abstract

The implementation of protocol P on a sending host S decides, through protocol P's routing mechanism, that it wants to transmit to a target host T located some place on a connected piece of 10Mbit Ethernet cable. To actually transmit the Ethernet packet a 48.bit Ethernet address must be generated. The addresses of hosts within protocol P are not always compatible with the corresponding Ethernet address (being different lengths or values). Presented here is a protocol that allows dynamic distribution of the information needed to build tables to translate an address A in protocol P's address space into a 48.bit Ethernet address.

Generalizations have been made which allow the protocol to be used for non-10Mbit Ethernet hardware. Some packet radio networks are examples of such hardware.

The protocol proposed here is the result of a great deal of discussion with several other people, most notably J. Noel Chiappa, Yogen Dalal, and James E. Kulp, and helpful comments from David Moon.

[The purpose of this RFC is to present a method of Converting Protocol Addresses (e.g., IP addresses) to Local Network Addresses (e.g., Ethernet addresses). This is a issue of general concern in the ARPA Internet community at this time. The method proposed here is presented for your consideration and comment. This is not the specification of a Internet Standard.]

Notes:

This protocol was originally designed for the DEC/Intel/Xerox 10Mbit Ethernet. It has been generalized to allow it to be used for other types of networks. Much of the discussion will be directed toward the 10Mbit Ethernet. Generalizations, where applicable, will follow the Ethernet-specific discussion.

DOD Internet Protocol will be referred to as Internet.

Numbers here are in the Ethernet standard, which is high byte first. This is the opposite of the byte addressing of machines such as PDP-11s and VAXes. Therefore, special care must be taken with the opcode field (ar\$op) described below.

An agreed upon authority is needed to manage hardware name space values (see below). Until an official authority exists, requests should be submitted to

David C. Plummer
Symbolics, Inc.
243 Vassar Street
Cambridge, Massachusetts 02139

Alternatively, network mail can be sent to DCP@MIT-MC.

The Problem:

The world is a jungle in general, and the networking game contributes many animals. At nearly every layer of a network architecture there are several potential protocols that could be used. For example, at a high level, there is TELNET and SUPDUP for remote login. Somewhere below that there is a reliable byte stream protocol, which might be CHAOS protocol, DOD TCP, Xerox BSP or DECnet. Even closer to the hardware is the logical transport layer, which might be CHAOS, DOD Internet, Xerox PUP, or DECnet. The 10Mbit Ethernet allows all of these protocols (and more) to coexist on a single cable by means of a type field in the Ethernet packet header. However, the 10Mbit Ethernet requires 48.bit addresses on the physical cable, yet most protocol addresses are not 48.bits long, nor do they necessarily have any relationship to the 48.bit Ethernet address of the hardware. For example, CHAOS addresses are 16.bits, DOD Internet addresses are 32.bits, and Xerox PUP addresses are 8.bits. A protocol is needed to dynamically distribute the correspondences between a <protocol, address> pair and a 48.bit Ethernet address.

Motivation:

Use of the 10Mbit Ethernet is increasing as more manufacturers supply interfaces that conform to the specification published by DEC, Intel and Xerox. With this increasing availability, more and more software is being written for these interfaces. There are two alternatives: (1) Every implementor invents his/her own method to do some form of address resolution, or (2) every implementor uses a standard so that his/her code can be distributed to other systems without need for modification. This proposal attempts to set the standard.

Definitions:

Define the following for referring to the values put in the TYPE field of the Ethernet packet header:

- ether_type\$XEROX_PUP,
- ether_type\$DOD_INTERNET,
- ether_type\$CHAOS,

and a new one:

- ether_type\$ADDRESS_RESOLUTION.

Also define the following values (to be discussed later):

- ares_op\$REQUEST (= 1, high byte transmitted first) and

- ares_op\$REPLY (= 2),

and

- ares_hrd\$Ethernet (= 1).

Packet format:

To communicate mappings from <protocol, address> pairs to 48.bit Ethernet addresses, a packet format that embodies the Address Resolution protocol is needed. The format of the packet follows.

Ethernet transmission layer (not necessarily accessible to the user):

- 48.bit: Ethernet address of destination

- 48.bit: Ethernet address of sender

- 16.bit: Protocol type = ether_type\$ADDRESS_RESOLUTION

Ethernet packet data:

- 16.bit: (ar\$hrd) Hardware address space (e.g., Ethernet, Packet Radio Net.)

- 16.bit: (ar\$pro) Protocol address space. For Ethernet hardware, this is from the set of type fields ether_typ\$<protocol>.

- 8.bit: (ar\$hln) byte length of each hardware address

- 8.bit: (ar\$pln) byte length of each protocol address

- 16.bit: (ar\$op) opcode (ares_op\$REQUEST | ares_op\$REPLY)

- nbytes: (ar\$sha) Hardware address of sender of this packet, n from the ar\$hln field.

- mbytes: (ar\$spa) Protocol address of sender of this packet, m from the ar\$pln field.

- nbytes: (ar\$tha) Hardware address of target of this packet (if known).

- mbytes: (ar\$tpa) Protocol address of target.

Packet Generation:

As a packet is sent down through the network layers, routing determines the protocol address of the next hop for the packet and on which piece of hardware it expects to find the station with the immediate target protocol address. In the case of the 10Mbit Ethernet, address resolution is needed and some lower layer (probably the hardware driver) must consult the Address Resolution module (perhaps implemented in the Ethernet support module) to convert the <protocol type, target protocol address> pair to a 48.bit Ethernet address. The Address Resolution module tries to find this pair in a table. If it finds the pair, it gives the corresponding 48.bit Ethernet address back to the caller (hardware driver) which then transmits the packet. If it does not, it probably informs the caller that it is throwing the packet away (on the assumption the packet will be retransmitted by a higher network layer), and generates an Ethernet packet with a type field of ether_type\$ADDRESS_RESOLUTION. The Address Resolution module then sets the ar\$hrd field to ares_hrd\$Ethernet, ar\$pro to the protocol type that is being resolved, ar\$hln to 6 (the number of bytes in a 48.bit Ethernet address), ar\$pln to the length of an address in that protocol, ar\$op to ares_op\$REQUEST, ar\$sha with the 48.bit ethernet address of itself, ar\$spa with the protocol address of itself, and ar\$tpa with the protocol address of the machine that is trying to be accessed. It does not set ar\$tha to anything in particular, because it is this value that it is trying to determine. It could set ar\$tha to the broadcast address for the hardware (all ones in the case of the 10Mbit Ethernet) if that makes it convenient for some aspect of the implementation. It then causes this packet to be broadcast to all stations on the Ethernet cable originally determined by the routing mechanism.

Packet Reception:

When an address resolution packet is received, the receiving Ethernet module gives the packet to the Address Resolution module which goes through an algorithm similar to the following. Negative conditionals indicate an end of processing and a discarding of the packet.

?Do I have the hardware type in ar\$hrd?

Yes: (almost definitely)

[optionally check the hardware length ar\$hln]

?Do I speak the protocol in ar\$pro?

Yes:

[optionally check the protocol length ar\$pln]

Merge_flag := false

If the pair <protocol type, sender protocol address> is already in my translation table, update the sender hardware address field of the entry with the new information in the packet and set Merge_flag to true.

?Am I the target protocol address?

Yes:

If Merge_flag is false, add the triplet <protocol type, sender protocol address, sender hardware address> to the translation table.

?Is the opcode ares_op\$REQUEST? (NOW look at the opcode!!)

Yes:

Swap hardware and protocol fields, putting the local hardware and protocol addresses in the sender fields.

Set the ar\$op field to ares_op\$REPLY

Send the packet to the (new) target hardware address on the same hardware on which the request was received.

Notice that the <protocol type, sender protocol address, sender hardware address> triplet is merged into the table before the opcode is looked at. This is on the assumption that communication is bidirectional; if A has some reason to talk to B, then B will probably have some reason to talk to A. Notice also that if an entry already exists for the <protocol type, sender protocol address> pair, then the new hardware address supersedes the old one. Related Issues gives some motivation for this.

Generalization: The ar\$hrd and ar\$hln fields allow this protocol and packet format to be used for non-10Mbit Ethernets. For the 10Mbit Ethernet <ar\$hrd, ar\$hln> takes on the value <1, 6>. For other hardware networks, the ar\$pro field may no longer correspond to the Ethernet type field, but it should be associated with the protocol whose address resolution is being sought.

Why is it done this way??

Periodic broadcasting is definitely not desired. Imagine 100 workstations on a single Ethernet, each broadcasting address resolution information once per 10 minutes (as one possible set of parameters). This is one packet every 6 seconds. This is almost reasonable, but what use is it? The workstations aren't generally going to be talking to each other (and therefore have 100 useless entries in a table); they will be mainly talking to a mainframe, file server or bridge, but only to a small number of other workstations (for interactive conversations, for example). The protocol described in this paper distributes information as it is needed, and only once (probably) per boot of a machine.

This format does not allow for more than one resolution to be done in the same packet. This is for simplicity. If things were multiplexed the packet format would be considerably harder to digest, and much of the information could be gratuitous. Think of a bridge that talks four protocols telling a workstation all four protocol addresses, three of which the workstation will probably never use.

This format allows the packet buffer to be reused if a reply is generated; a reply has the same length as a request, and several of the fields are the same.

The value of the hardware field (ar\$hrd) is taken from a list for this purpose. Currently the only defined value is for the 10Mbit Ethernet (ares_hrd\$Ethernet = 1). There has been talk of using this protocol for Packet Radio Networks as well, and this will require another value as will other future hardware mediums that wish to use this protocol.

For the 10Mbit Ethernet, the value in the protocol field (ar\$pro) is taken from the set ether_type\$. This is a natural reuse of the assigned protocol types. Combining this with the opcode (ar\$op) would effectively halve the number of protocols that can be resolved under this protocol and would make a monitor/debugger more complex (see Network Monitoring and Debugging below). It is hoped that we will never see 32768 protocols, but Murphy made some laws which don't allow us to make this assumption.

In theory, the length fields (ar\$hlen and ar\$pln) are redundant, since the length of a protocol address should be determined by the hardware type (found in ar\$hrd) and the protocol type (found in ar\$pro). It is included for optional consistency checking, and for network monitoring and debugging (see below).

The opcode is to determine if this is a request (which may cause a reply) or a reply to a previous request. 16 bits for this is overkill, but a flag (field) is needed.

The sender hardware address and sender protocol address are absolutely necessary. It is these fields that get put in a translation table.

The target protocol address is necessary in the request form of the packet so that a machine can determine whether or not to enter the sender information in a table or to send a reply. It is not necessarily needed in the reply form if one assumes a reply is only provoked by a request. It is included for completeness, network monitoring, and to simplify the suggested processing algorithm described above (which does not look at the opcode until AFTER putting the sender information in a table).

The target hardware address is included for completeness and network monitoring. It has no meaning in the request form, since it is this number that the machine is requesting. Its meaning in the reply form is the address of the machine making the request. In some implementations (which do not get to look at the 14 byte ethernet header, for example) this may save some register shuffling or stack space by sending this field to the hardware driver as the hardware destination address of the packet.

There are no padding bytes between addresses. The packet data should be viewed as a byte stream in which only 3 byte pairs are defined to be words (ar\$hrd, ar\$pro and ar\$op) which are sent most significant byte first (Ethernet/PDP-10 byte style).

Network monitoring and debugging:

The above Address Resolution protocol allows a machine to gain knowledge about the higher level protocol activity (e.g., CHAOS, Internet, PUP, DECnet) on an Ethernet cable. It can determine which Ethernet protocol type fields are in use (by value) and the protocol addresses within each protocol type. In fact, it is not necessary for the monitor to speak any of the higher level protocols involved. It goes something like this:

When a monitor receives an Address Resolution packet, it always enters the <protocol type, sender protocol address, sender hardware address> in a table. It can determine the length of the hardware and protocol address from the ar\$hlen and ar\$plen fields of the packet. If the opcode is a REPLY the monitor can then throw the packet away. If the opcode is a REQUEST and the target protocol address matches the protocol address of the monitor, the monitor sends a REPLY as it normally would. The monitor will only get one mapping this way, since the REPLY to the REQUEST will be sent directly to the requesting host. The monitor could try sending its own REQUEST, but this could get two monitors into a REQUEST sending loop, and care must be taken.

Because the protocol and opcode are not combined into one field, the monitor does not need to know which request opcode is associated with which reply opcode for the same higher level protocol. The length fields should also give enough information to enable it to "parse" a protocol addresses, although it has no knowledge of what the protocol addresses mean.

A working implementation of the Address Resolution protocol can also be used to debug a non-working implementation. Presumably a hardware driver will successfully broadcast a packet with Ethernet type field of ether_type\$ADDRESS_RESOLUTION. The format of the packet may not be totally correct, because initial implementations may have bugs, and table management may be slightly tricky. Because requests are broadcast a monitor will receive the packet and can display it for debugging if desired.

An Example:

Let there exist machines X and Y that are on the same 10Mbit Ethernet cable. They have Ethernet address EA(X) and EA(Y) and DOD Internet addresses IPA(X) and IPA(Y). Let the Ethernet type of Internet be ET(IP). Machine X has just been started, and sooner or later wants to send an Internet packet to machine Y on the same cable. X knows that it wants to send to IPA(Y) and tells the hardware driver (here an Ethernet driver) IPA(Y). The driver consults the Address Resolution module to convert <ET(IP), IPA(Y)> into a 48.bit Ethernet address, but because X was just started, it does not have this information. It throws the Internet packet away and instead creates an ADDRESS RESOLUTION packet with

```
(ar$hrd) = ares_hrd$Ethernet
(ar$pro) = ET(IP)
(ar$hln) = length(EA(X))
(ar$pln) = length(IPA(X))
(ar$op)  = ares_op$REQUEST
(ar$sha) = EA(X)
(ar$spa) = IPA(X)
(ar$tha) = don't care
(ar$tpa) = IPA(Y)
```

and broadcasts this packet to everybody on the cable.

Machine Y gets this packet, and determines that it understands the hardware type (Ethernet), that it speaks the indicated protocol (Internet) and that the packet is for it ((ar\$tpa)=IPA(Y)). It enters (probably replacing any existing entry) the information that <ET(IP), IPA(X)> maps to EA(X). It then notices that it is a request, so it swaps fields, putting EA(Y) in the new sender Ethernet address field (ar\$sha), sets the opcode to reply, and sends the packet directly (not broadcast) to EA(X). At this point Y knows how to send to X, but X still doesn't know how to send to Y.

Machine X gets the reply packet from Y, forms the map from <ET(IP), IPA(Y)> to EA(Y), notices the packet is a reply and throws it away. The next time X's Internet module tries to send a packet to Y on the Ethernet, the translation will succeed, and the packet will (hopefully) arrive. If Y's Internet module then wants to talk to X, this will also succeed since Y has remembered the information from X's request for Address Resolution.

Related issue:

It may be desirable to have table aging and/or timeouts. The implementation of these is outside the scope of this protocol. Here is a more detailed description (thanks to MOON@SCRC@MIT-MC).

If a host moves, any connections initiated by that host will work, assuming its own address resolution table is cleared when it moves. However, connections initiated to it by other hosts will have no particular reason to know to discard their old address. However, 48.bit Ethernet addresses are supposed to be unique and fixed for all time, so they shouldn't change. A host could "move" if a host name (and address in some other protocol) were reassigned to a different physical piece of hardware. Also, as we know from experience, there is always the danger of incorrect routing information accidentally getting transmitted through hardware or software error; it should not be allowed to persist forever. Perhaps failure to initiate a connection should inform the Address Resolution module to delete the information on the basis that the host is not reachable, possibly because it is down or the old translation is no longer valid. Or perhaps receiving of a packet from a host should reset a timeout in the address resolution entry used for transmitting packets to that host; if no packets are received from a host for a suitable length of time, the address resolution entry is forgotten. This may cause extra overhead to scan the table for each incoming packet. Perhaps a hash or index can make this faster.

The suggested algorithm for receiving address resolution packets tries to lessen the time it takes for recovery if a host does move. Recall that if the <protocol type, sender protocol address> is already in the translation table, then the sender hardware address supersedes the existing entry. Therefore, on a perfect Ethernet where a broadcast REQUEST reaches all stations on the cable, each station will be get the new hardware address.

Another alternative is to have a daemon perform the timeouts. After a suitable time, the daemon considers removing an entry. It first sends (with a small number of retransmissions if needed) an address resolution packet with opcode REQUEST directly to the Ethernet address in the table. If a REPLY is not seen in a short amount of time, the entry is deleted. The request is sent directly so as not to bother every station on the Ethernet. Just forgetting entries will likely cause useful information to be forgotten, which must be regained.

Since hosts don't transmit information about anyone other than themselves, rebooting a host will cause its address mapping table to be up to date. Bad information can't persist forever by being passed around from machine to machine; the only bad information that can exist is in a machine that doesn't know that some other machine has changed its 48.bit Ethernet address. Perhaps manually resetting (or clearing) the address mapping table will suffice.

This issue clearly needs more thought if it is believed to be important. It is caused by any address resolution-like protocol.

Network Working Group
Request for Comments: 1048

P. Prindeville
McGill University
February 1988

BOOTP Vendor Information Extensions

Status of this Memo

This memo proposes an addition to the Bootstrap Protocol (BOOTP). Comments and suggestions for improvements are sought. Distribution of this memo is unlimited.

Introduction

As workstations and personal computers proliferate on the Internet, the administrative complexity of maintaining a network is increased by an order of magnitude. The assignment of local network resources to each client represents one such difficulty. In most environments, delegating such responsibility to the user is not plausible and, indeed, the solution is to define the resources in uniform terms, and to automate their assignment.

The basic Bootstrap Protocol [RFC-951] dealt with the issue of assigning an internet address to a client, as well as a few other resources. The protocol included provisions for vendor-defined resource information.

This memo defines a (potentially) vendor-independent interpretation of this resource information.

Overview of BOOTP

While the Reverse Address Resolution (RARP) Protocol [RFC-903] may be used to assign an IP address to a local network hardware address, it provides only part of the functionality needed. Though this protocol can be used in conjunction with other supplemental protocols (the Resource Location Protocol [RFC-887], the Domain Name System [RFC-883]), a more integrated solution may be desirable.

Bootstrap Protocol (BOOTP) is a UDP/IP-based protocol that allows a booting host to configure itself dynamically, and more significantly, without user supervision. It provides a means to assign a host its IP address, a file from which to download a boot program from some server, that server's address, and (if present) the address of an Internet gateway.

One obvious advantage of this procedure is the centralized management of network addresses, which eliminates the need for per-host unique configuration files. In an environment with several hundred hosts, maintaining local configuration information and operating system versions specific to each host might otherwise become chaotic. By categorizing hosts into classes and maintaining configuration information and boot programs for each class, the complexity of this chore may be reduced in magnitude.

BOOTP Vendor Information Format

The full description of the BOOTP request/reply packet format may be found in [RFC-951]. The rest of this document will concern itself with the last field of the packet, a 64 octet area reserved for vendor information, to be used in a hitherto unspecified fashion. A generalized use of this area for giving information useful to a wide class of machines, operating systems, and configurations follows. In situations where a single BOOTP server is to be used among heterogeneous clients in a single site, a generic class of data may be used.

Vendor Information "Magic Cookie"

As suggested in [RFC-951], the first four bytes of this field have been assigned to the magic cookie, which identifies the mode in which the succeeding data is to be interpreted. The value of the magic cookie is the 4 octet dotted decimal 99.130.83.99 (or hexadecimal number 63.82.53.63) in network byte order.

Format of Individual Fields

The vendor information field has been implemented as a free format, with extendable tagged sub-fields. These sub-fields are length tagged (with exceptions; see below), allowing clients not implementing certain types to correctly skip fields they cannot interpret. Lengths are exclusive of the tag and length octets; all multi-byte quantities are in network byte-order.

Fixed Length Data

The fixed length data are comprised of two formats. Those that have no data consist of a single tag octet and are implicitly of one-octet length, while those that contain data consist of one tag octet, one length octet, and length octets of data.

Pad Field (Tag: 0, Data: None)

May be used to align subsequent fields to word boundaries

required by the target machine (i.e., 32-bit quantities such as IP addresses on 32-bit boundaries).

Subnet Mask Field (Tag: 1, Data: 4 subnet mask bytes)

Specifies the net and local subnet mask as per the standard on subnetting [RFC-950]. For convenience, this field must precede the GATEWAY field (below), if present.

Time Offset Field (Tag: 2, Data: 4 time offset bytes)

Specifies the time offset of the local subnet in seconds from Coordinated Universal Time (UTC); signed 32-bit integer.

End Field (Tag: 255, Data: None)

Specifies end of usable data in the vendor information area. The rest of this field should be filled with PAD zero octets.

Variable Length Data

The variable length data has a single format; it consists of one tag octet, one length octet, and length octets of data.

Gateway Field (Tag: 3, Data: N address bytes)

Specifies the IP addresses of N/4 gateways for this subnet. If one of many gateways is preferred, that should be first.

Time Server Field (Tag: 4, Data: N address bytes)

Specifies the IP addresses of N/4 time servers [RFC-868].

IEN-116 Name Server Field (Tag: 5, Data: N address bytes)

Specifies the IP addresses of N/4 name servers [IEN-116].

Domain Name Server Field (Tag: 6, Data: N address bytes)

Specifies the IP addresses of N/4 domain name servers RFC-883].

Log Server Field (Tag: 7, Data: N address bytes)

Specifies the IP addresses of N/4 MIT-LCS UDP log server [LOGGING].

Cookie/Quote Server Field (Tag: 8, Data: N address bytes)

Specifies the IP addresses of N/4 Quote of the Day servers [RFC-865].

LPR Server Field (Tag: 9, Data: N address bytes)

Specifies the IP addresses of N/4 Berkeley 4BSD printer servers [LPD].

Impress Server Field (Tag: 10, Data: N address bytes)

Specifies the IP addresses of N/4 Impress network image servers [IMAGEN].

RLP Server Field (Tag: 11, Data: N address bytes)

Specifies the IP addresses of N/4 Resource Location Protocol (RLP) servers [RFC-887].

Hostname (Tag: 12, Data: N bytes of hostname)

Specifies the name of the client. The name may or may not domain qualified: this is a site-specific issue.

Reserved Fields (Tag: 128-254, Data: N bytes of undefined content)

Specifies additional site-specific information, to be interpreted on an implementation-specific basis. This should follow all data with the preceding generic tags 0-127).

Extensions

Additional generic data fields may be registered by contacting:

Joyce K. Reynolds
USC - Information Sciences Institute
4676 Admiralty Way
Marina del Rey, California 90292-6695

or by E-mail as: JKREYNOLDS@ISI.EDU
(nic handle JKRL).

Implementation specific use of undefined generic types (those in the range 12-127) may conflict with other implementations, and registration is required.

When selecting information to put into the vendor specific area, care should be taken to not exceed the 64 byte length restriction. Nonessential information (such as host name and quote of the day server) may be excluded, which may later be located with a more appropriate service protocol, such as RLP or the WKS resource-type of the domain name system. Indeed, even RLP servers may be discovered using a broadcast request to locate a local RLP server.

Comparison to Alternative Approaches

Extending BOOTP to provide more configuration information than the minimum required by boot PROMs may not be necessary. Rather than having each module in a host (e.g., the time module, the print spooler, the domain name resolver) broadcast to the BOOTP server to obtain the addresses of required servers, it would be better for each of them to multicast directly to the particular server group of interest, possibly using "expanding ring" multicasts.

The multicast approach has the following advantages over the BOOTP approach:

- It eliminates dependency on a third party (the BOOTP server) that may be temporarily unavailable or whose database may be incorrect or incomplete. Multicasting directly to the desired services will locate those servers that are currently available, and only those.
- It reduces the administrative chore of keeping the (probably replicated) BOOTP database up-to-date and consistent. This is especially important in an environment with a growing number of services and an evolving population of servers.
- In some cases, it reduces the amount of packet traffic and/or the delay required to get the desired information. For example, the current time can be obtained by a single multicast to a time server group which evokes replies from those time servers that are currently up. The BOOTP approach would require a broadcast to the BOOTP server, a reply from the BOOTP server, one or more unicasts to time servers (perhaps waiting for long timeouts if the initially chosen server(s) are down), and finally a reply from a server.

One apparent advantage of the proposed BOOTP extensions is that they provide a uniform way to locate servers. However, the multicast approach could also be implemented in a consistent way across multiple services. The V System naming protocol is a good example of this; character string pathnames are used to name any number of resources (i.e., not just files) and a standard subroutine library looks after multicasting to locate the resources, caching the discovered locations, and detecting stale cache data.

Another apparent advantage of the BOOTP approach is that it allows an administrator to easily control which hosts use which servers. The multicast approach favors more distributed control over resource allocation, where each server decides which hosts it will serve, using whatever level of authentication is appropriate for the particular service. For example, time servers usually don't care who they serve (i.e., administrative control via the BOOTP database is unnecessary), whereas file servers usually require strong authentication (i.e., administrative control via the BOOTP database is insufficient).

The main drawback of the multicast approach, of course, is that IP multicasting is not widely implemented, and there is a need to locate existing services which do not understand IP multicasts.

The BOOTP approach may be most efficient in the case that all the information needed by the client host is returned by a single BOOTP reply and each program module simply reads the information it needs from a local table filled in by the BOOTP reply.

Acknowledgments

I would like to thank the following persons for their helpful comments and insights into this memo: Drew Perkins, of Carnegie Mellon University, Bill Croft, of Stanford University, and co-author of BOOTP, and Steve Deering, also of Stanford University, for contributing the "Comparison to Alternative Approaches" section.

References

- [RFC-951] Croft, B., and J. Gilmore, "Bootstrap Protocol", Network Information Center, SRI International, Menlo Park, California, September 1985.
- [RFC-903] Finlayson, R., T. Mann, J. Mogul, and M. Theimer, "A Reverse Address Resolution Protocol", Network Information Center, SRI International, Menlo Park, California, June 1984.
- [RFC-887] Accetta, M., "Resource Location Protocol", Network Information Center, SRI International, Menlo Park, California, December 1983.
- [RFC-883] Mockapetris, P., "Domain Name - Implementation and Specification", Network Information Center, SRI International, Menlo Park, California, November 1983.
- [RFC-950] Mogul, J., "Internet Standard Subnetting Procedure",

Network Information Center, SRI International, Menlo Park, California, August 1985.

- [RFC-868] Postel, J., "Time Protocol", Network Information Center, SRI International, Menlo Park, California, May 1983.
- [IEN-116] Postel, J., "Internet Name Server", Network Information Center, SRI International, Menlo Park, California, August 1979.
- [LOGGING] Clark, D., "Logging and Status Protocol", Massachusetts Institute of Technology Laboratory for Computer Science, Cambridge, Massachusetts, 1981.
- [RFC-865] Postel, J., "Quote of the Day Protocol", Network Information Center, SRI International, Menlo Park, California, May 1983.
- [LPD] Campbell, R., "4.2BSD Line Printer Spooler Manual", UNIX Programmer's Manual, Vol II, University of California at Berkeley, Computer Science Division, July 1983.
- [IMAGEN] "Image Server XT Programmer's Guide", Imagen Corporation, Santa Clara, California, August 1986.